（第Ⅰ卷）

潘凯华 刘中华 著

PHP开发实战1200例



清華大学出版社
TSINGHUA UNIVERSITY PRESS

软件开发实战1200例

PHP开发实战1200例

（第I卷）

潘凯华　刘中华　等编著

清华大学出版社

北　京

内 容 简 介

《PHP 开发实战 1200 例》包括第Ⅰ卷、第Ⅱ卷共计 1200 个例子，本书是第Ⅰ卷，共计 625 个例子。

本书以开发人员在项目开发中经常遇到的问题和必须掌握的技术为中心，介绍了应用 PHP 进行 Web 开发的各个方面的知识和技巧，主要包括开发环境、PHP 基础、Web 页面交互、文件操作、会话应用、图形图像处理及面向对象等内容。全书分为 5 篇 15 章，共计 625 个实例和 625 个经验技巧。每个实例都经作者精心筛选，具有很强的实用性，其中一些实例是开发人员难于寻觅的解决方案。

本书两卷共计 1200 个例子，包括了开发中各个方面最常用的实例，是目前市场上实例最全面的开发类图书；书中实例来源于多位工程师的多年积累，具有很强的实用性。

本书附带有配套 DVD 光盘，光盘中提供有部分实例的视频和大部分源代码，读者稍加修改即可应用。

本书适合 PHP 的初学者，如高校学生、求职人员作为练习、速查、学习使用，也适合 PHP 程序员参考、查阅。

图书在版编目（CIP）数据

PHP 开发实战 1200 例（第Ⅰ卷）/潘凯华，刘中华等编著．—北京：清华大学出版社，2011.1
（软件开发实战 1200 例）

ISBN 978-7-302-24334-2

Ⅰ. ①P…　Ⅱ. ①潘…　②刘…　Ⅲ. ①PHP 语言－程序设计　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2010）第 238611 号

责任编辑： 刘利民　毛姗姗
版式设计： 杨　洋
责任校对： 姜　彦
责任印制：
出版发行： 清华大学出版社　　**地　　址：** 北京清华大学学研大厦 A 座
http://www.tup.com.cn　　**邮　　编：** 100084
社 总 机： 010-62770175　　**邮　　购：** 010-62786544
投稿与读者服务： 010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈： 010-62772015，zhiliang@tup.tsinghua.edu.cn
印 刷 者：
装 订 者：
经　　销： 全国新华书店
开　　本： 203×260　**印　　张：** 58　**字　　数：** 1915 千字
（附 DVD 视频光盘 1 张）
版　　次： 2011 年 1 月第 1 版　**印　　次：** 2011 年 1 月第 1 次印刷
印　　数： 1～4000
定　　价： 96.00 元

产品编号：040557-01

前　言

Preface

特别说明：

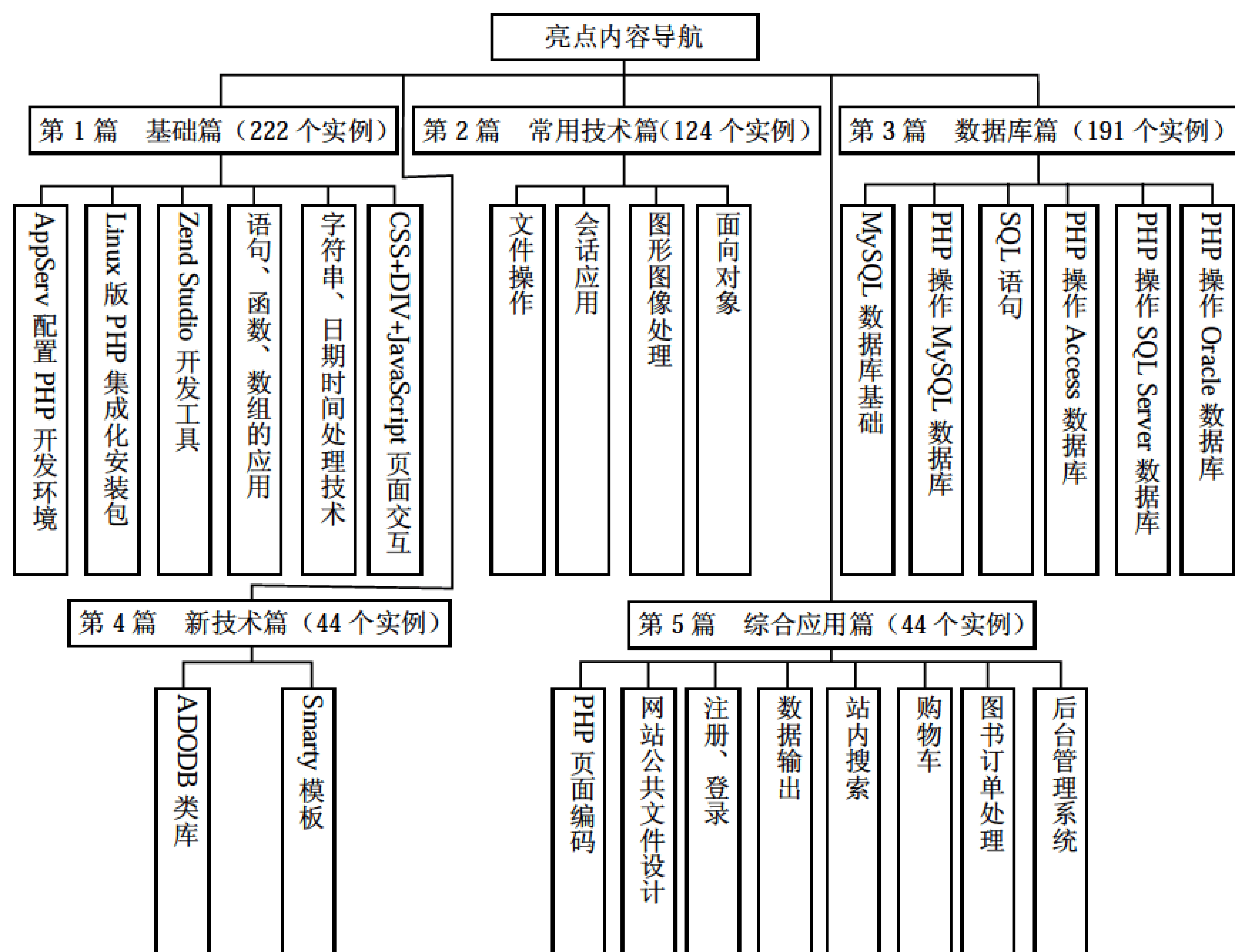
《PHP 开发实战 1200 例》包括第Ⅰ卷、第Ⅱ卷共计 1200 个例子，本书是第Ⅰ卷，包含 625 个例子。

程序开发是一项艰辛的工作，它不仅考验开发人员的智力，更考验开发人员的体力。纵观国内的软件企业，内部开发团队挑灯夜战、连续加班是常有的事情。笔者曾有幸参加过十几个项目的开发工作，对编程有深刻的体会。编程是一项复杂的创造性工作，不仅需要开发人员掌握各方面的知识，还需要开发人员具有丰富的开发经验。项目开发中的一个问题可能会占用团队几天甚至十几天的时间，但是如果开发人员遇到过类似的问题，也许几分钟就可以解决。这就是编程经验的重要性，也是许多软件企业招聘时选择有项目开发经验人员的主要原因。

本书精选项目开发中的 625 个实例，涵盖 PHP 基础、Web 页面交互、文件操作、面向对象、MySQL 数据库、ADODB 类库和 Smarty 模板等方面的内容，每一个知识点都提供有丰富的、具有很强实用性的实例，帮助开发人员快速解决疑难问题。

本书内容

本书分为 5 篇 15 章内容，共计 625 个实例。书中所选实例均来源于一线开发人员的实际开发项目，涵盖开发中经常使用和需要解决的热点、难点问题，可以使读者快速地解决开发中的难题，提高编程效率。本书亮点内容导航如下图所示。



为了方便读者学习书中的实例，积累开发经验，本书在讲解实例时按照实例说明、关键技术、设计过程和秘笈心法 4 个阶段进行。

- ☑ 实例说明

  实例说明部分利用图文结合的方式，介绍实例的特点、功能和运行效果，使读者快速了解实例的作用、可以从实例中学到的技能。
- ☑ 关键技术

  关键技术部分剖析实例使用的核心技术。这样，读者不必查看设计过程和源代码就可以快速掌握实例的关键技术，甚至直接做出实例。
- ☑ 设计过程

  设计过程部分详细介绍实例的开发过程。读者可以按照该部分内容做出自己的实例程序。
- ☑ 秘笈心法

  秘笈心法部分给出与实例相关的技巧、注意事项或经验总结，丰富读者的编程经验。

下图展示了一个完整实例的文档缩略图。

实例 275　通过图像显示密码安全强度　　中级

光盘位置：光盘\MR\06\275　　趣味指数：★★★☆

■ 实例说明

在本实例中，将介绍图像的另一项妙用——显示注册填写密码的安全强度。运行本实例，当在密码文本框中输入一个密码时，光标离开后，将自动判断这个密码的安全强度，并以图像的形式返回判断结果，其运行结果如图6.6所示。

图 6.6　通过图像显示密码安全强度

■ 关键技术

对密码的安全强度进行判断是在 JavaScript 脚本中，应用正则表达式来完成的，其原理如下：

（1）对密码中的值进行判断，如果只包含数字、英文字符串或者特殊字符串中的任意一种，那么说明密码安全强度差。

（2）对密码中的值进行判断，如果包含数字、英文字符串或者特殊字符串中的任意两种，那么说明密码安全强度中。

（3）对密码中的值进行判断，如果同时包含数字、英文字符串和特殊字符串，那么说明密码安全强度强。

■ 设计过程

本实例完成一个用户注册的功能，并且在表单中通过 JavaScript 脚本对提交的元素进行判断，最终符合要求后，将用户注册信息添加到指定的数据表中。

（1）创建 index.php 页面，完成用户注册页面和表单的设计，通过 JavaScript 对表单中提交的值进行判断，并直接将判断的结果反馈到 div 标签中，最终将表单中的数据提交到 index_ok.php 文件中，完成用户注册的操作。

```
<form name="form_reg" method="post" action="index_ok.php" onSubmit="return chkreginfo(form_reg,'all')">
<tr>
    <td height="30"><div align="right">密码：</div></td>
    <td height="30" colspan="2" align="left"> 
<input type="password" name="pwd" id="pwd" size="30" class="inputcss"
                onBlur="chkreginfo(form_reg,1)"> <font color="#FF0000">*</font> 
        <div id="chknew_pwd" style="color: #FF0000"></div>
    </td>
</tr>
</form>
```

（2）编写 JavaScript 脚本，创建 check.js 脚本文件，完成对表单中提交的值进行判断。

（3）创建 index_ok.php 文件，获取表单中提交的数据，将数据添加到指定的数据表中。

（4）创建 conn.php 文件，完成 MySQL 服务器和 MySQL 数据库的连接操作，并设置编码格式为utf-8。

■ 秘笈心法

心法领悟 275：正则表达式的运用。

在本实例中，不只一次使用了正则表达式。分别通过正则表达式对密码的强度、手机号码、座机号码以及邮箱地址进行判断。

```
/^13(\d{9})$|^15(\d{9})$|^189(\d{8})$/          //判断手机号码
/^(\d{3}-)(\d{8})$|^(\d{4}-)(\d{7})$|^(\d{4}-)(\d{8})$/ //判断座机号
/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/        //判断邮箱
```

本书特点

☑ **实例极为丰富**

本书第Ⅰ卷、第Ⅱ卷共精选了 1200 个实例和 1200 个经验技巧，涵盖了编程中多个方面的各种应用。用户开发中所需的技术、技巧在本书几乎都可以找到。

☑ **例子非常实用**

书中所选实例均是项目开发中经常需要使用或实现的技术，由数名工程师多年积累而成，许多实例是开发人员梦寐以求的。

☑ **可操作性很强**

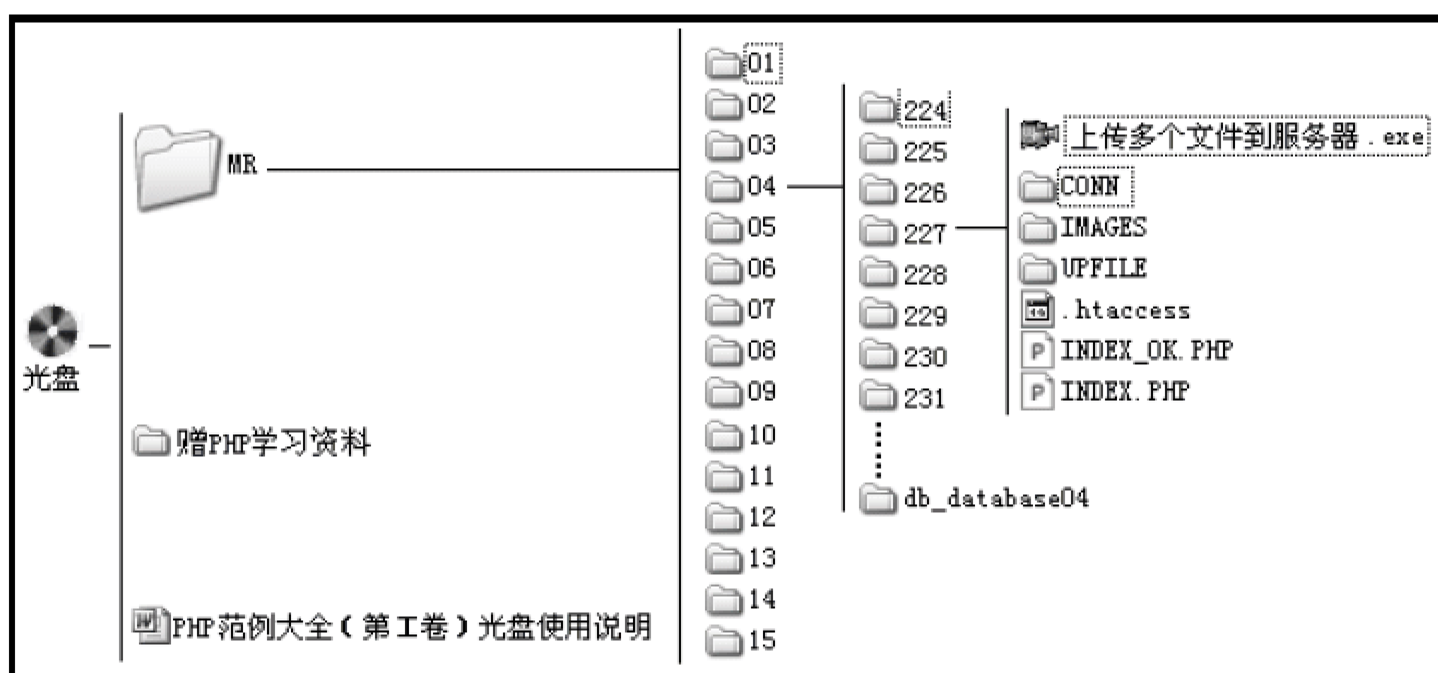
本书实例都给出了程序说明、设计思路、设计步骤、源程序等内容，读者可按部就班地直接使用。

☑ **完善的技术服务**

为了帮助读者尽快理解内容，读者朋友可通过 QQ、BBS、邮箱、电话等多种形式获取相关问题的答复。

本书 DVD 光盘

为了帮助读者学习和使用书中的实例，本书附赠有 DVD 光盘，其中不仅提供了书中所有实例项目的源代码、素材，还提供了关键实例的开发视频讲解及 PHP 学习资料。光盘目录如下图所示。



本书约定

☑ **实例使用方法**

用户在学习本书的过程中，可以从光盘中复制实例，去掉实例的只读属性。有些实例需要使用相应的数据库或第三方资源，这些实例在使用前需要进行相应配置，详细使用方式请参考本书的光盘使用说明书。此外，如果用户直接将本书实例用于商业用途，由此产生的不良后果由用户自己承担。

☑ **实例及录像位置**

书中实例在光盘中的存储格式为“MR\章号\实例序号”。例如，“MR\04\096”表示实例 096，位于第 4 章。书中关键实例在光盘中提供了开发录像，录像的存储格式与实例的存储格式是相同的。例如，查看实例 096 的开发录像，可以在光盘中的“MR\04\096”路径下找到，录像名称与实例名称相同，格式为 EXE 格式。

☑ **录像使用说明**

本书赠送的实例录像采用 EXE 文件格式，无须使用播放器，双击就可以直接播放。在主目录中，目录前有录像图标的实例，表示在光盘中存在视频录像。

☑ **部分实例只给出关键代码**

由于篇幅限制，书中有些实例只给出了关键代码，完整代码请参考光盘实例程序。

读者人群

本书非常适合以下人员阅读：

☑ PHP 初学者，如高校大学生、求职人员、培训机构学员

☑ PHP 程序员

学习答疑

如果您在学习或使用本书的过程中遇到问题或疑惑，可以通过如下方式与我们联系。

☑ 服务网站：www.mingribook.com

☑ 服务 QQ：100310265 或 100310063

☑ 服务信箱：mingrisoft@mingrisoft.com

☑ 服务电话：0431-84978981/84978982

☑ 学习社区：www.mrbccd.com

我们承诺将在 5 个工作日内给您提供解答。

本书作者

本书由明日科技组织编写，参加编写的程序员有潘凯华、刘中华、杨明、刘欣、李慧、董大永、李继业、尹强、张磊、赛奎春、高春艳、宋坤、刘锐宁、梁水、李伟明、刘彬彬、安剑、孙秀梅、巩建华、房大伟、吕双、刘云峰、王国辉、李钟尉、张振坤、陈丹丹、赵会东、卢翰、白伟明、王小科、梁晓岚、杨丽、顾丽丽、刘龄龄、张彦国、王乐乐、陈英、刘莹、李贺、李鑫等。由于作者水平有限，疏漏和不足之处在所难免，请广大读者朋友批评指正。

编　者

目　录

Contents

第 1 篇　基础篇

第 2 篇 常用技术篇

第 3 篇　数据库篇

第 4 篇　新技术篇

第 5 篇　综合应用篇

说明：《PHP 开发实战 1200 例（第 II 卷）》将于近期出版，出版日期请留意 www.mingribook.com 相关公告，其目录如下。

《PHP 开发实战 1200 例（第 II 卷）》目录

第 1 篇　Web 应用篇

第 2 篇　网络应用篇

第 3 篇　数据库与抽象层篇

第 5 篇 框架与项目整合篇

第1篇

基础篇

第 1 章

开发环境

- AppServ——PHP 集成化安装包
- XAMPP——PHP 集成化安装包
- IIS+PHP+MySQL——独立搭建 PHP 开发环境
- LAMP——独立配置 PHP 开发环境
- XAMPP——Linux 版 PHP 集成化安装包
- Dreamweaver 开发工具
- Zend Studio 开发工具

1.1　AppServ——PHP 集成化安装包

AppServ 将 Apache、PHP、MySQL 和 phpMyAdmin 等服务器软件和工具安装配置完成后进行打包处理，开发人员只要到网站上下载该软件，然后安装，即可完成 PHP 开发环境的快速搭建，非常适合初学者使用。

实例 001	通过 AppServ 配置 PHP 开发环境 光盘位置：光盘\MR\01\001	初级 趣味指数：★★★★

实例说明

在本实例中，讲解如何通过 AppServ 配置 PHP 的开发环境，配置完成后将输出如图 1.1 所示的页面。

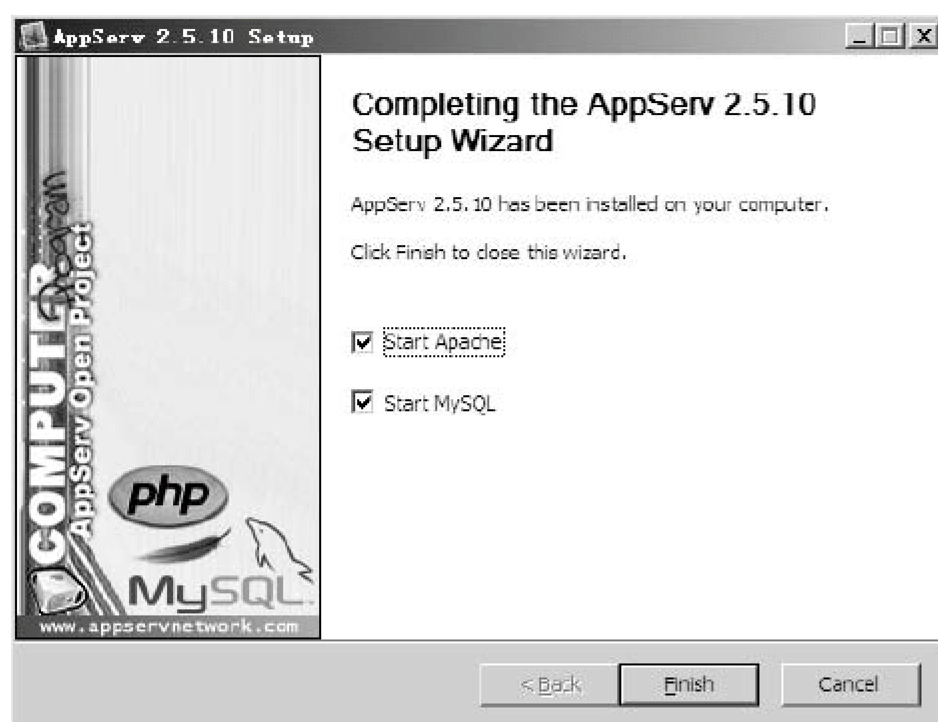


图 1.1　PHP 开发环境配置完成

关键技术

通过 AppServ 配置 PHP 开发环境时，首先从互联网上下载 AppServ 的安装文件，下载地址为 http://www.AppServnetwork.com/。

注意：在使用 AppServ 搭建 PHP 开发环境时，必须确保在系统中没有安装 Apache、PHP 和 MySQL。否则，要先将这些软件卸载，然后应用 AppServ。

在软件的安装过程中，要重点注意下面几个步骤：

☑ 设置安装路径；
☑ 选择安装软件；
☑ 设置端口号；
☑ 设置数据库密码；
☑ 设置数据库编码格式。

设计过程

下面对应用 AppServ 集成化安装包搭建 PHP 开发环境的具体操作步骤进行讲解。

（1）双击 AppServ-win32-2.5.10.exe 文件，打开如图 1.2 所示的 AppServ 启动页面。

（2）单击图 1.2 中的 Next 按钮，打开如图 1.3 所示的 AppServ 安装协议页面。



图 1.2　AppServ 启动页面



图 1.3　AppServ 安装协议

（3）单击图 1.3 中的 I Agree 按钮，打开如图 1.4 所示的页面。设置 AppServ 的安装路径（默认安装路径一般为 C:\AppServ），AppServ 安装完成后，Apache、MySQL、PHP 都将以子目录的形式存储到该目录下。

（4）单击图 1.4 中的 Next 按钮，打开如图 1.5 所示的页面，选择要安装的程序和组件（默认为全选）。

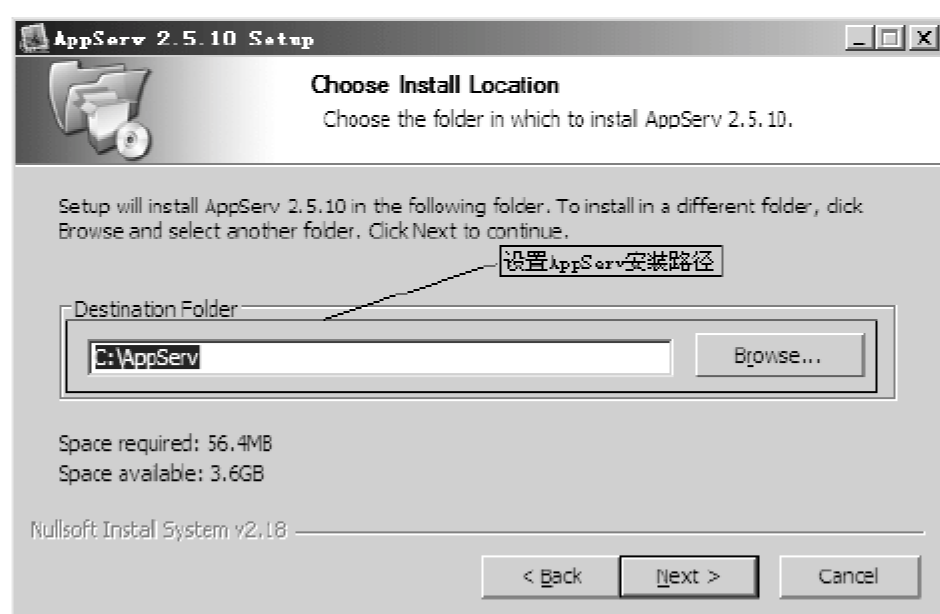


图 1.4　AppServ 安装路径选择

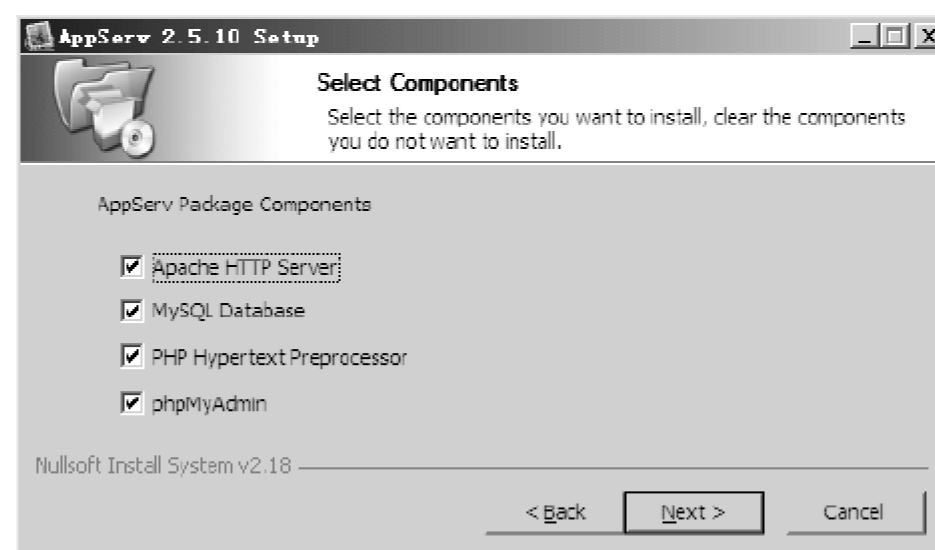


图 1.5　AppServ 安装选项

技巧： 在第（4）步中，如果本机已经安装 MySQL 数据库，那么可以不勾选 MySQL Database 选项，仍使用本机已经存在的 MySQL 数据库。

（5）在图 1.5 中单击 Next 按钮，打开如图 1.6 所示的页面。填写计算机名称、添加邮箱地址、设置 Apache 的端口号（默认为 80 端口）。

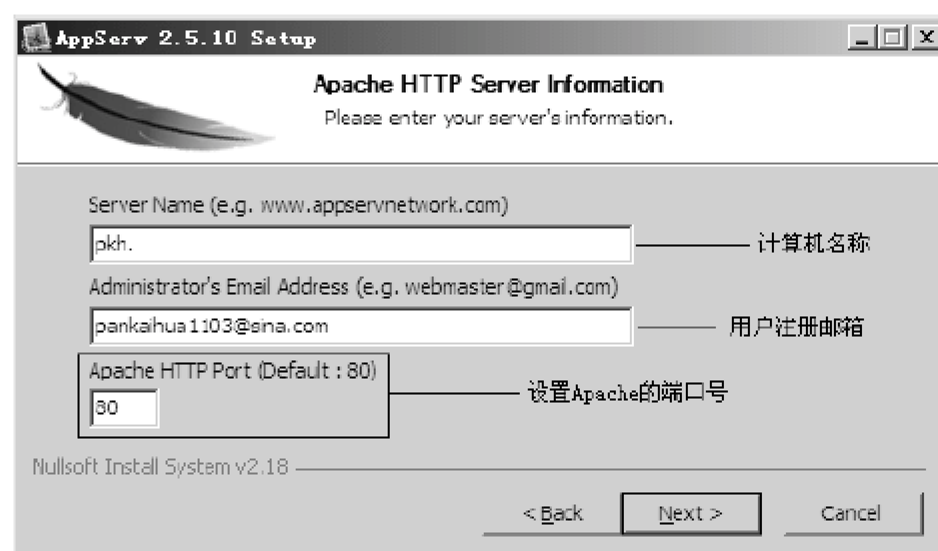


图 1.6　Apache 端口号设置

注意： Apache 服务器端口号的设置，直接关系到 Apache 服务器是否能够正常启动。如果本机中的 80 端口被 IIS 或者迅雷占用，那么这里就需要修改 Apache 的端口号，或者修改 IIS 和迅雷的端口号，才能完成 Apache 服务器的配置。如果出现端口冲突，那么将导致安装失败，Apache 服务不能启动。

（6）单击图 1.6 中的 Next 按钮，打开如图 1.7 所示的页面。设置 MySQL 数据库 root 用户的登录密码及字符集。

说明：MySQL 数据库的字符集，可以设置为 UTF-8、GBK 或者 GB2312 格式。这里将字符集设置为“GB2312 Simplified Chinese”，表示 MySQL 数据库的字符集将采用简体中文形式。



图 1.7　设置 MySQL 数据库 root 用户的密码和字符集

技巧：对于在图 1.7 中设置的 MySQL 数据库 root 用户的密码必须牢记，因为程序在连接数据库时必须使用这个密码。建议读者将这个密码设置为 111，因为这是在开发本书中的程序时所使用的数据库密码。如此，在运行本书中的数据库程序时，就不需要修改密码，否则要修改连接数据库的密码。如果忘记安装时设置的密码，最直接有效的解决方式是重新安装 AppServ。

（7）单击图 1.7 中的 Install 按钮，开始安装，如图 1.8 所示。

（8）安装完成后可以在开始菜单的 AppServ 相关操作列表中启动 Apache 及 MySQL 服务，如图 1.9 所示。

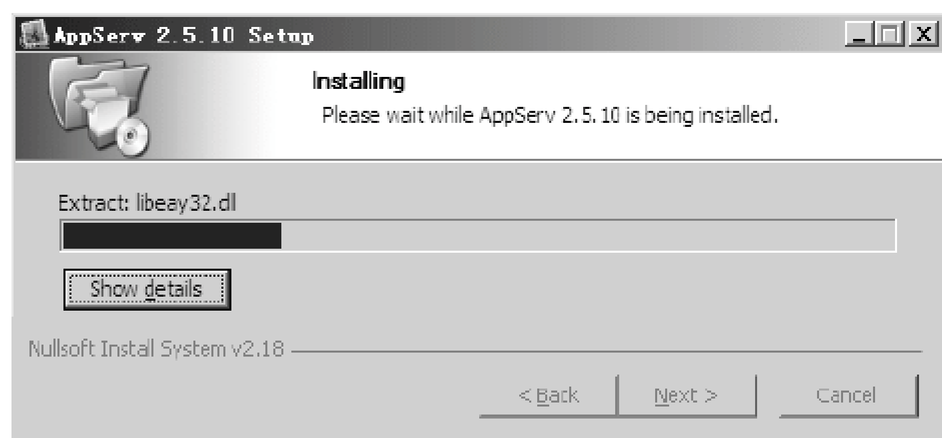


图 1.8　AppServ 安装页面

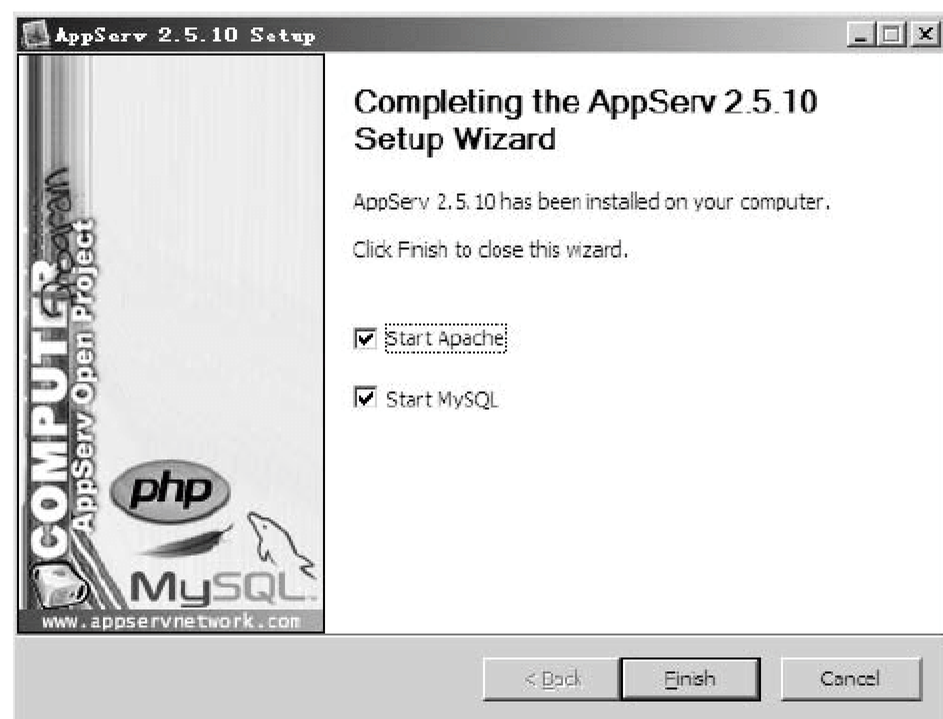


图 1.9　AppServ 安装完成页面

秘笈心法

心法领悟 001：Apache 的配置文件。

Apache 的配置文件存储于 C:\AppServ\Apache2.2\conf\httpd.conf。通过 Apache 配置文件可以设置 Apache 的端口号、Apache 服务器的根目录或者控制 Apache 是否支持伪静态。

（1）设置 Apache 的端口号：

```
Listen 80
```

（2）设置 Apache 服务器的根目录：

```
DocumentRoot "C:/AppServ/www"
```

（3）设置对伪静态的支持：

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

将该项前面的“#”去掉，可启动该项。查找 httpd.conf 文件，找到其中的“AllowOverride”项，将其值都修改为 All。

完成对 Apache 的修改后，必须保存文件，并重新启动 Apache 服务器才能够生效。

实例 002 AppServ 应用技巧

光盘位置：光盘\MR\01\002

中级

趣味指数：★★★★☆

实例说明

安装 AppServ 后，可以对其进行测试。在 IE 浏览器的地址栏中输入 hppt://127.0.0.1，如果输出如图 1.10 所示的页面，则说明 PHP 开发环境配置成功。



图 1.10　PHP 开发环境配置成功

关键技术

AppServ 安装完成后，整个目录默认安装在 C:\AppServ，此目录下包含 4 个子目录，如图 1.11 所示，用户可以将所有程序文件存储到 www 目录下。

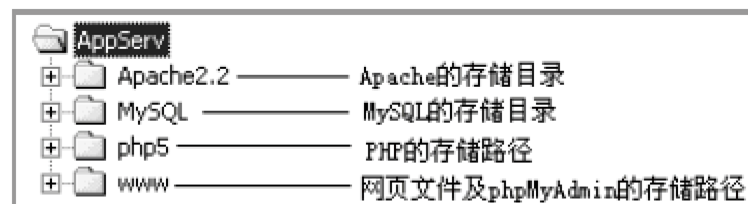


图 1.11　AppServ 目录结构

在 Apache2.2\conf\目录下，有一个 httpd.conf 文件，它是 Apache 服务器的配置文件，在这个文件中可以修改 Apache 服务器的端口号、根目录等，所有有关 Apache 服务器的配置都在这个文件中完成。

在 MySQL 目录下，有一个 my.ini 文件，它是 MySQL 服务器的配置文件，存储 MySQL 的配置信息。

在 MySQL\data 目录下存储的是数据库文件，所有程序使用的数据库都存储在这个文件夹下。

说明： 如果 MySQL 服务器的版本是 5.1，那么数据库文件夹 data 的默认存储位置是 C:/Documents and Settings/All Users/Application Data/MySQL/MySQL Server 5.1/Data/。

在 php5\ext 文件夹下存储的是 PHP 内置的函数类库，以.dll 的格式存储。而 PHP 的配置文件 php.ini 存储在本机系统盘的 windows 文件夹下。

www 目录是程序运行的根目录，也就是说我们要运行的所有程序都必须存储在这个目录下。phpMyAdmin

图形化管理工具默认存储在此目录下。

设计过程

（1）www 文件夹存储网站的运行文件，其默认的内容如图 1.12 所示。



图 1.12　www 文件夹的默认内容

（2）其中，index.php 是默认访问 127.0.0.1 时浏览的内容；而 phpinfo.php 存储的是 PHP 的配置信息和支持的模块。phpinfo.php 的内容如图 1.13 所示。

PHP Version 5.3.1

System	Windows NT PKH 5.2 build 3790 (Windows Server 2003 Standard Edition) i586
Build Date	Nov 20 2009 17:20:57
Compiler	MSVC6 (Visual C++ 6.0)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled

图 1.13　phpinfo.php 的内容

（3）其中的 phpMyAdmin 文件夹存储的是 phpMyAdmin 图形化管理工具。

秘笈心法

心法领悟 002：AppServ 的相关说明。

（1）如果是应用 AppServ 配置的 PHP 开发环境，那么在 IE 浏览器中直接输入 127.0.0.1 将输出如图 1.10 所示的页面，因为在 www 文件夹下有一个默认的 index.php 文件；如果要运行 www 文件夹下的其他文件，那么必须在 127.0.0.1 后指定具体的文件夹或者文件，例如，输入 127.0.0.1/text.php 或者 127.0.0.1/mr/。

（2）Apache 的启动和停止

单击“开始”按钮，选择“程序”/AppServ/Control Server by Service 命令，在弹出的菜单中完成 Apache、MySQL 服务器的启动和停止操作。

单击“开始”按钮，选择“设置”/“控制面板”/“管理工具”/“服务”命令，找到 Apache2.2，单击右键，完成 Apache 的启动、停止或者重新启动操作。

完成对 php.ini 文件的修改后，同样要保存文件，重新启动 Apache 服务器后，修改才能够生效。

实例 003	第 1 个 PHP 程序 光盘位置：光盘\MR\01\003	初级 趣味指数：★★★★

实例说明

AppServ 安装、调试成功后，开发第 1 个 PHP 程序，输出系统的当前时间，运行结果如图 1.14 所示。



图 1.14　第 1 个 PHP 程序

关键技术

（1）应用 PHP 中的 date 函数，输出系统的当前时间，设置时间的格式为年-月-日 时：分：秒（Y-m-d H:i:s）。

（2）将编写的 index.php 文件存储于 www 文件夹下的 MR\01\003 目录下。

在运行本实例时，会发现程序输出的时间与系统的当前时间相差了 8 个小时，至于为什么会出现此种情况，我们将在本实例的秘笈心法中讲解。

设计过程

（1）通过 Dreamweaver 创建一个 index.php 脚本文件，存储于 MR/01/003/文件夹下。

（2）应用 date()函数输出系统的当前时间，其代码如下：

```
<?php
echo date("Y-m-d H:i:s");                                    //输出系统的当前时间
?>
```

（3）在 IE 浏览器中输入 http://127.0.0.1/mr/01/003/，运行结果如图 1.14 所示。

秘笈心法

心法领悟 003：对于系统时间的相关说明。

程序输出时间与系统当前时间出现偏差的关键是 php.ini 文件的设置。在 php.ini 文件中，默认设置的是标准的格林威治时间，即采用的是英国伦敦时间，而我们的当地时间是北京时间，所以会出现时间的偏差。

PHP 的配置文件存储于 C:\WINDOWS\php.ini 文件中。在 php.ini 文件中，定位到[date]下的“;date.timezone =”选项，去掉前面的分号，并设置它的值为当地所在时区使用的时间，即可统一程序输出的时间与系统当前的时间。

```
;date.timezone =
```

例如，如果当地所在时区为东八区，那么就可以设置 date.timezone 的值为 PRC、Asia/Hong_Kong、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐）等，这些都是东八区的时间。

1.2　XAMPP——PHP 集成化安装包

XAMPP（Apache+MySQL+PHP+PERL）是一个功能强大的建站集成软件包，可以在 Windows、Linux、Solaris 3 种操作系统下安装使用，支持多种语言，包括英文、简体中文、繁体中文、韩文、俄文、日文等。

实例 004	通过 XAMPP 配置 PHP 开发环境 光盘位置：光盘\MR\01\004	初级 趣味指数：★★★★☆

实例说明

XAMPP 是一个易于安装且包含 MySQL、PHP 和 Perl 的 Apache 发行版。安装成功后的运行效果如图 1.15 所示。

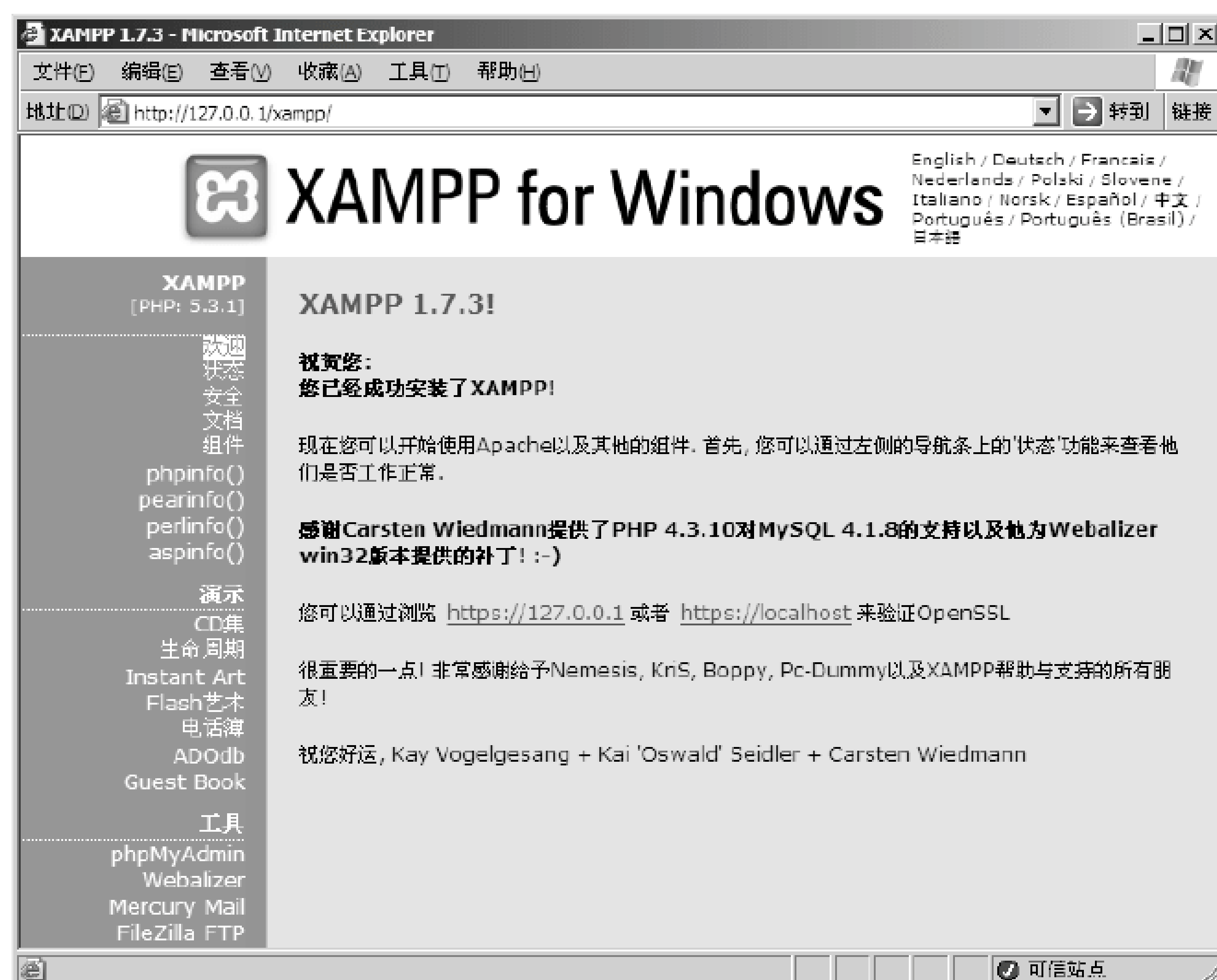


图 1.15　XAMPP 安装成功

关键技术

目前 XAMPP 的 Windows 版本是 1.7.3，包括如下内容：

☑　Apache 2.2.14 (IPv6 enabled)
☑　MySQL 5.1.41 + PBXT engine
☑　PHP 5.3.1
☑　OpenSSL 0.9.8l
☑　phpMyAdmin 3.2.4
☑　XAMPP Control Panel 2.5.8
☑　XAMPP CLI Bundle 1.6
☑　Webalizer 2.21-02
☑　Mercury Mail Transport System v4.72
☑　msmtp 1.4.19
☑　FileZilla FTP Server 0.9.33
☑　SQLite 2.8.17
☑　SQLite 3.6.20
☑　ADOdb 5.10
☑　eAccelerator 0.9.6-rc1
☑　Xdebug 2.0.6-dev
☑　Ming 0.4.3

适用于 Windows 2000/XP/2003/Vista/7。

在 Windows 操作系统下，XAMPP 有两种形式：

☑　自解压 RAR 包：简单且安全。
☑　ZIP 压缩包：XAMPP 的普通 ZIP 压缩档案。

设计过程

XAMPP 1.7.3 版的具体安装步骤如下：

（1）运行.exe 安装文件，将弹出如图 1.16 所示的操作界面，在这里设置 XAMPP 的安装路径。

（2）安装路径设置完成后，单击 Install 按钮，执行 XAMPP 的安装，如图 1.17 所示。

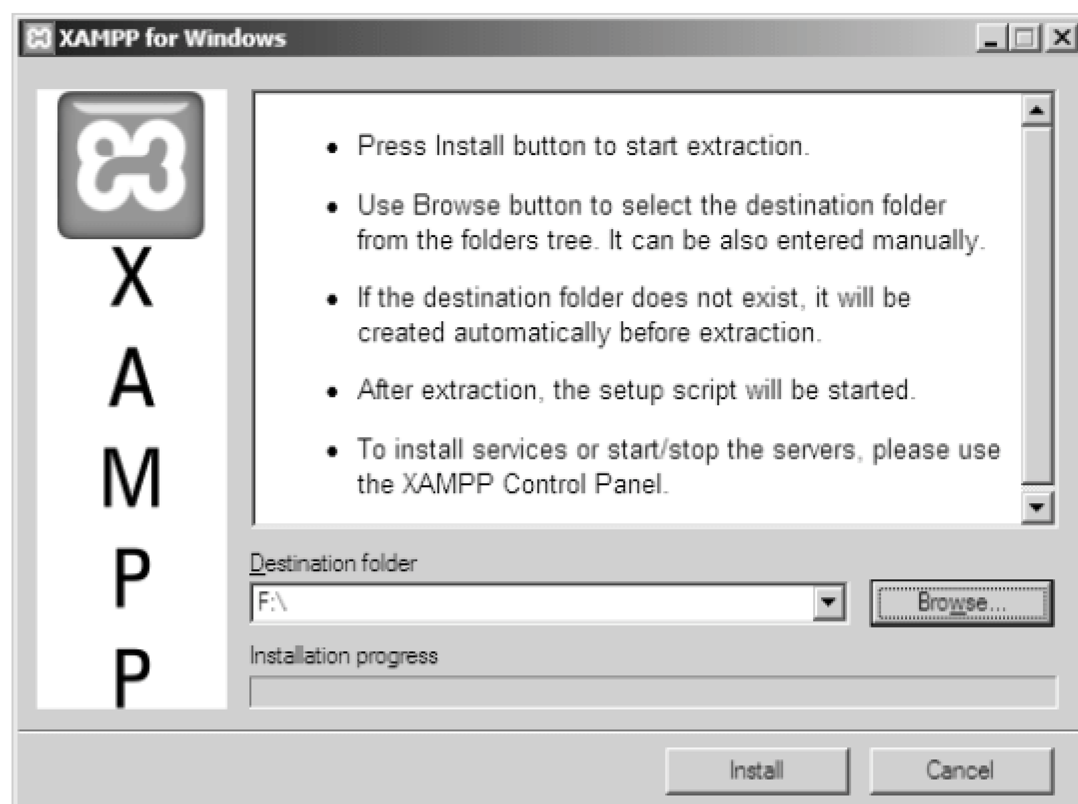


图 1.16　设置 XAMPP 的安装路径

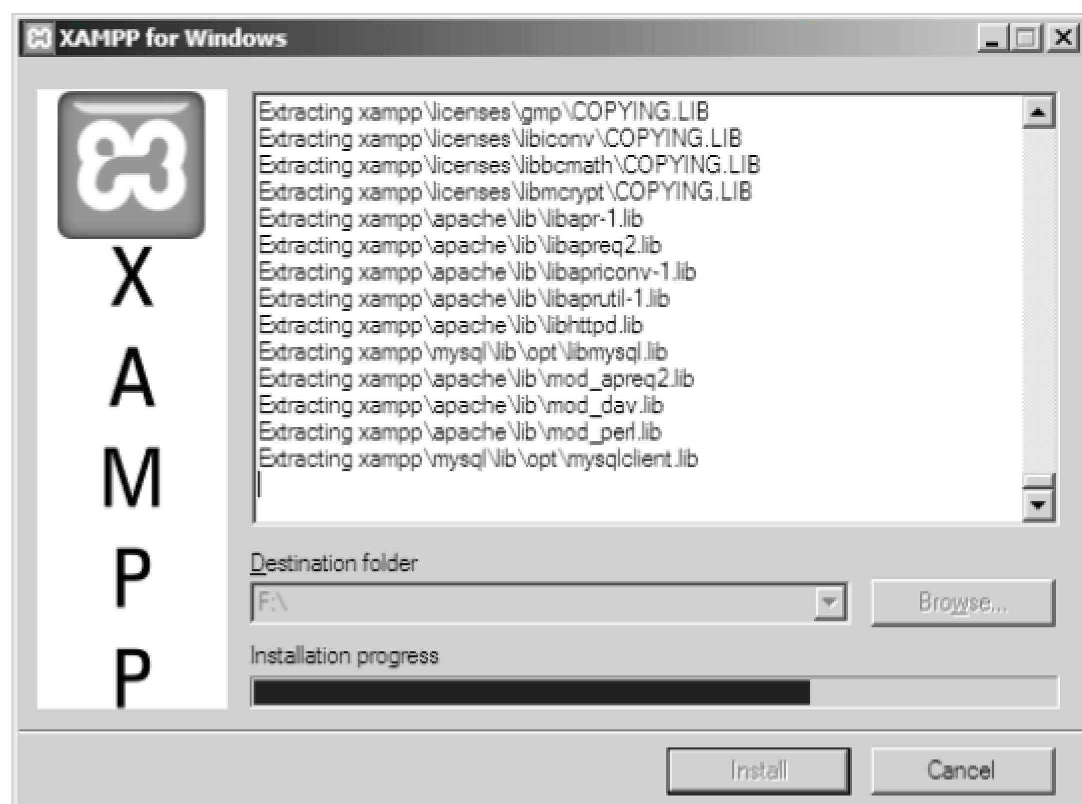


图 1.17　安装 XAMPP

（3）XAMPP 安装成功后，将弹出如图 1.18 所示的操作界面，在其中对 XAMPP 进行设置，包括是否创建桌面快捷方式、是否在开机时启动 XAMPP 等。

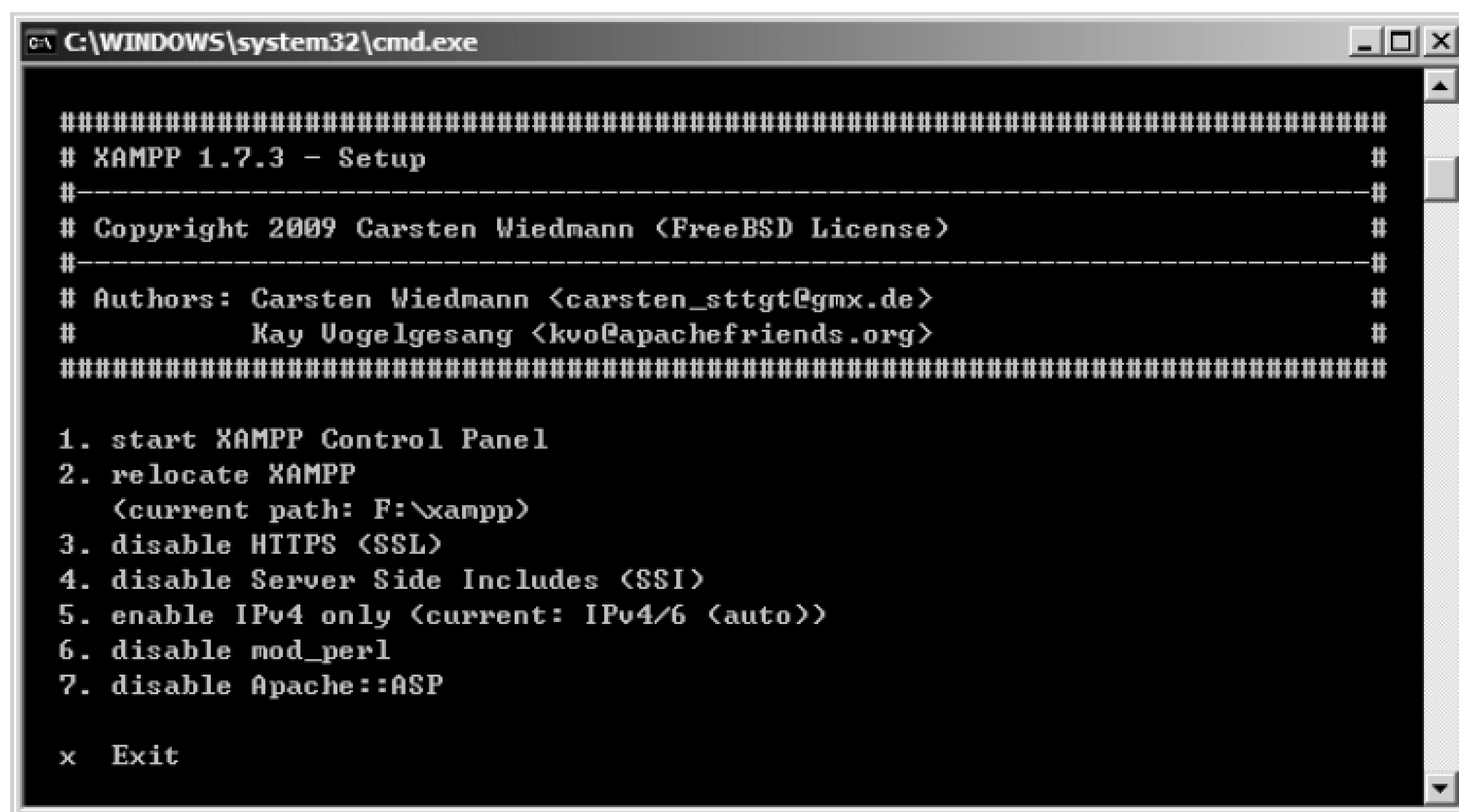


图 1.18　设置 XAMPP

至此，XAMPP 的安装、配置完毕。

秘笈心法

心法领悟 004：XAMPP 中的配置文件。

XAMPP 中各种配置文件的存储位置如下：

☑ Apache 基本配置：.\xampp\apache\conf\httpd.conf

☑ Apache SSL：.\xampp\apache\conf\ssl.conf

☑ Apache Perl（仅限插件）：.\xampp\apache\conf\perl.conf

☑ PHP：.\xampp\php\php.ini

☑ MySQL：.\xampp\mysql\bin\my.ini

☑ phpMyAdmin：.\xampp\phpMyAdmin\config.inc.php

☑ FileZilla FTP 服务器：.\xampp\FileZillaFTP\FileZilla

实例 005　测试 XAMPP 是否安装成功

光盘位置：光盘\MR\01\005

初级

趣味指数：★★★★☆

实例说明

成功安装 XAMPP 后，启动 Apache 服务器，在浏览器的地址栏中输入 http://localhost 或 http://127.0.0.1，即可检测所有的 XAMPP 样例和工具，运行结果如图 1.19 所示。

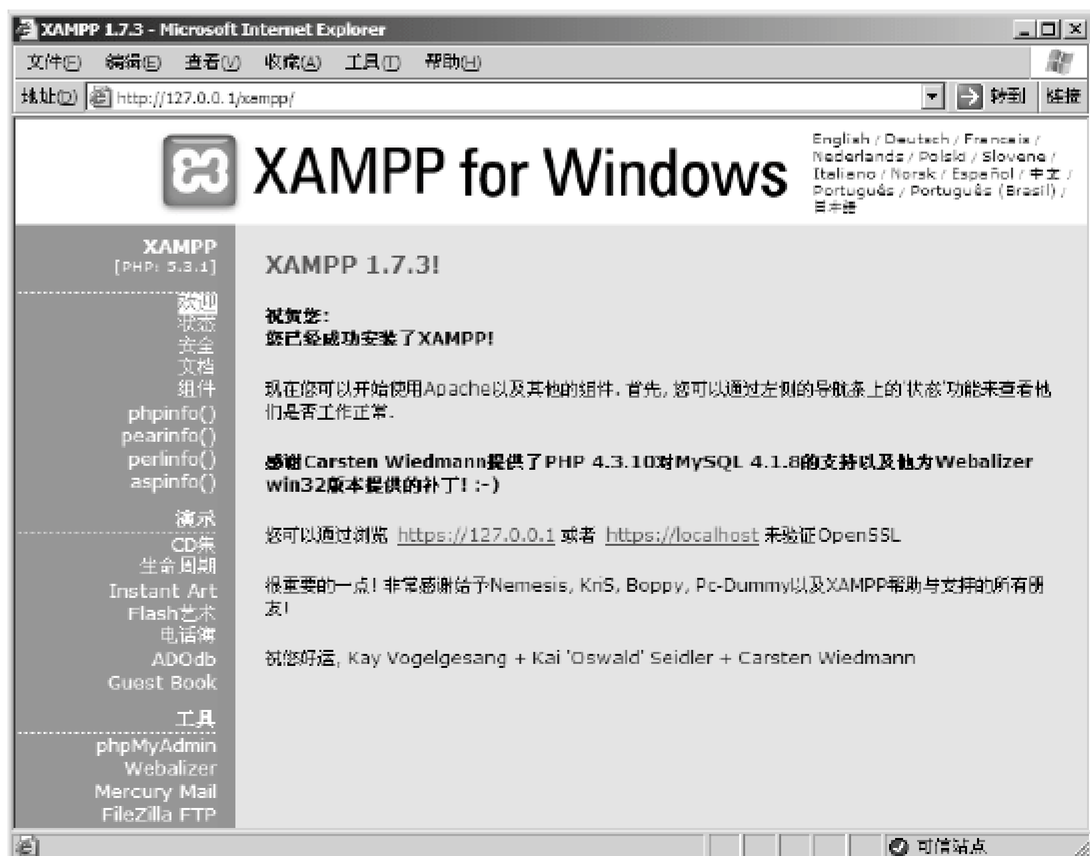


图 1.19　检测安装是否成功

关键技术

XAMPP 中默认将所有网络文档存储于 htdocs 主文件夹中（.\xampp\htdocs）。默认 htdocs 主文件夹下的内容如图 1.20 所示。



图 1.20　XAMPP 的默认网络文件

读者可以在浏览器的地址栏中输入 http://localhost/来检测 XAMPP 是否安装成功。

设计过程

（1）XAMPP 文件夹存储的是 XAMPP 的各种配置信息，当直接运行 index.php 文件时，调用的就是 XAMPP 文件夹下的 index.php 文件，输出如图 1.20 所示的内容。

（2）在 index.html 文件中，输出一个简单的测试信息，如图 1.21 所示。



图 1.21　通过 html 测试 PHP 是否安装成功

（3）在 index.php 文件中，通过 header()函数跳转到 xampp 文件夹下的 index.php，输出 XAMPP 的信息。

秘笈心法

心法领悟 005：如何启动、停止和测试 XAMPP。
通过文件\xampp\xampp-control.exe 控制 XAMPP 的启动、停止和测试。

实例 006 XAMPP 应用技巧

初级

光盘位置：光盘\MR\01\006

趣味指数：★★★★☆

实例说明

XAMPP 只为开发者服务，切记不要将其应用于生产环境，因为对于生产环境而言，使用 XAMPP 可能会带来灾难性的打击。

因为 MySQL 管理员（root）未设置密码，用户可以直接通过 phpMyAdmin 访问 MySQL 服务器，修改数据库中的数据，或者通过网络访问，运行结果如图 1.22 所示。



图 1.22　直接访问 MySQL 数据库

关键技术

要确保 MySQL 数据库的安全，必须为 MySQL 管理员（root）设置密码，其方法如下：

以 root 用户的身份登录 phpMyAdmin，在 phpMyAdmin 的权限中设置 root 用户的密码。操作界面如图 1.23 所示。



图 1.23　修改 MySQL 用户密码

说明： 更改 root 的密码之后，还要修改 phpMyAdmin 中的相关信息，具体方法请参看设计过程。

设计过程

在 phpMyAdmin 的权限中完成 root 用户的密码设置后，在\...\xampp\phpmyadmin\下搜索 config.inc.php 文件，定位到如下位置：

```
/* Authentication type and info */
$cfg['Servers'][$i]['auth_type']            = 'http';           //设置登录方式
$cfg['Servers'][$i]['user']                 = 'root';           //设置用户名
$cfg['Servers'][$i]['password']             = "111;             //设置密码
$cfg['Servers'][$i]['AllowNoPassword']      = true;
```

设置 auth_type 的值为“http”；设置 password 的值为“111”，这样登录 phpMyAdmin 时，就必须要输入密码，否则无法登录。

秘笈心法

心法领悟 006：修改 MySQL 缺省字符集。

修改 MySQL 缺省字符集为 utf8，定位到 my.ini 文件，需添加如下内容：

```
[client]
default_character_set=utf8
[mysqld]
# default-character-set=utf8
character-set-server = utf8
collation-server = utf8_general_ci
[mysql]
default_character_set=utf8
```

实例 007	第 2 个 PHP 程序 光盘位置：光盘\MR\01\007	初级 趣味指数：★★★★

实例说明

既然 XAMPP 只为开发者服务，那么我们就应用 XAMPP 开发一个 PHP 程序，熟悉一下 XAMPP 的应用。本实例简单地输出一段文字，其运行结果如图 1.24 所示。



图 1.24　第 2 个 PHP 程序

关键技术

（1）确定服务器的运行文件夹，编写 PHP 脚本文件，将其存储于 XAMPP 指定的运行文件夹下。

（2）应用 PHP 的 date()函数输出系统的当前时间，时间的格式是“年　月　日　时　分　秒”。

设计过程

（1）新建一个记事本文件，编写 PHP 脚本，通过 date()函数输出系统的当前时间，其代码如下：

```
<?php
echo "现在是：".date("Y 年 m 月 d 日  H 时 i 分 s 秒");
?>
```

（2）保存这个记事本文件，将其存储于 xampp\htdocs\MR\01\007（这是笔者在本机中定义的实例存储文件夹）。

秘笈心法

心法领悟 007：如何配置默认的开始页面？

如果在浏览器地址栏中输入 http://localhost/xampp/，Apache 服务器会自动返回一个默认的开始页面，Apache 会自动寻找已存在的类似 index.php 的索引页。httpd.conf 中的“DirectoryIndex”指令负责这个功能。在这里可以定义默认开始页面的名字和协议。

XAMPP 在默认状态下，“DirectoryIndex”指令的列表如下：

```
index.php index.php4 index.php3 index.cgi index.pl index.html index.htm index.html.var index.phtml
```

1.3 IIS+PHP+MySQL——独立搭建 PHP 开发环境

Internet 信息服务器即 Internet Information Server，缩写为 IIS，是 Microsoft 的 Web 服务器。它集成于 Windows NT Server 之中，方便易用，为 Web 应用程序提供了功能强大的运行平台。

本书使用 IIS 6.0，在 Windows 2003 操作系统下完成 PHP 环境的搭建，所使用的软件包括：

☑ PHP 软件：php-5.3.1-nts-Win32-VC9-x86

☑ MySQL 数据库：mysql-essential-5.1.39-win32.msi

说明：上述版本是笔者在开发本书时所使用的，读者可以根据自己的实际情况进行调整。如果是 Windows 7 操作系统，则使用 IIS 7.0，如果是 Windows XP 系统，则使用 IIS 5.0。

实例 008	安装 PHP 光盘位置：光盘\MR\01\008	初级 趣味指数：★★★★

实例说明

PHP 5.3.1 是 PHP 目前推出的最新版本，可以到 PHP 的官方网站下载，下载地址为 http：//www.php.net//downloads.php。

本实例讲解如何选择、安装 PHP 5.3.1。

关键技术

（1）版本的区分

目前推出的 PHP 5.3.1 有多个版本，包括 VC6、X86 和 VC9、X86。

VC6：legacy Visual Studio 6 compiler，是使用此编译器编译的。

VC9：the Visual Studio 2008 compiler，是用微软的 VS 编辑器编译的。

（2）版本的选择

如果在 Windows 下使用 Apache+PHP，选择 VC6 版本；

如果在 Windows 下使用 IIS+PHP，选择 VC9 版本。

（3）Non Thread Safe 和 Thread Safe

Non Thread Safe 是非线程安全；Thread Safe 是线程安全。官方不建议将 Non Thread Safe 应用于生产环境，所以我们选择 Thread Safe 版本的 PHP。

设计过程

下面讲解 php-5.3.1-nts-Win32-VC9-x86 的配置方法。

（1）将下载的文件 php-5.3.1-nts-Win32-VC9-x86.zip 解压，重新命名为 php，并将其复制到指定的目录下。例如 C:\php、F:\php 等。这里将 php 放置在 F:\PkhiisPHP 目录下。

（2）将 F:\PkhiisPHP\php 目录下的 libMySQL.dll 文件复制到 C:\windows\system32\（如果是 Windows 2000，则为 C:\WINNT\system32\）目录下。

（3）将 F:\PkhiisPHP\php 目录下的 php.ini-development 重命名为 php.ini，如图 1.25 所示，并将其复制到系统盘的 windows 文件夹下（例如，若 C 盘为系统盘，则是 C:\windows）。

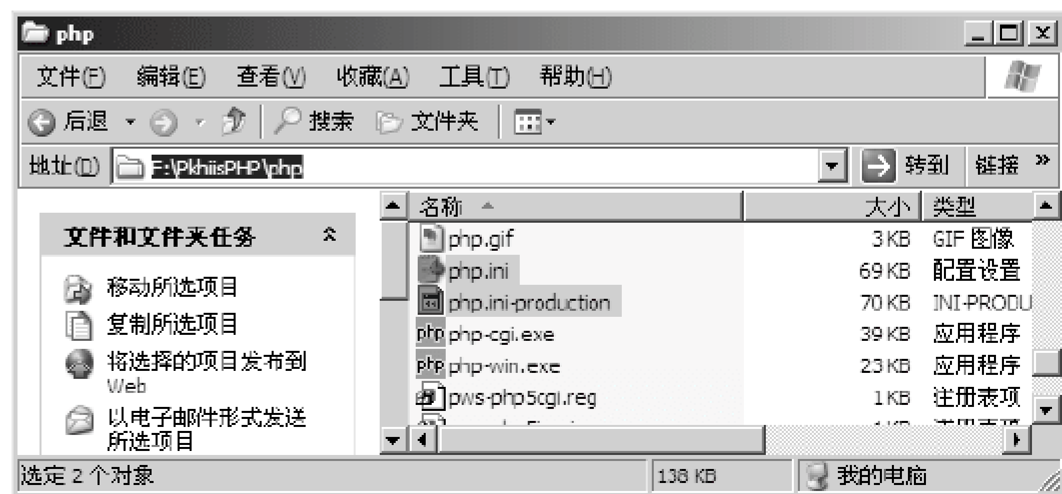


图 1.25　生成 php.ini

（4）打开 php.ini 文件并定位到 extension_dir = "./"这一行，修改为 extension_dir=" F:\PkhiisPHP\php\ext"。

（5）同样在 php.ini 文件中，将下面 5 项设置前的分号去掉，并设置相应的参数，完成对 CGI 和 FastCGI 的支持。

```
;cgi.force_redirect = 1
;cgi.fix_pathinfo=1
;fastcgi.impersonate = 1
;fastcgi.logging = 0
;cgi.rfc2616_headers = 1
```

（6）加载动态库。例如，定位到;extension=php_MySQL.dll 这一行，将前面的分号“;”去掉，完成 MySQL 动态库的加载。

（7）PHP 安装成功后，还要对 FastCGI 进行配置，在系统盘的\system32\inetsrv\文件夹下找到 fcgiext.ini 文件，在该文件的最后添加如下内容：

```
[Types]
php=PHP
[PHP]
ExePath=F:\PkhiisPHP\php\php-cgi.exe
```

其中，[Types]指定语言为 PHP；[PHP]指定 php-cgi.exe 在本机中的安装位置。

秘笈心法

心法领悟 008：使用 IIS 和 PHP 5.3.1 来配置 PHP 的开发环境。

如果使用 IIS 和 PHP 5.3.1 来配置 PHP 的开发环境，那么本机中必须安装微软的 VC++ 2008 的运行库（vcredist_x64.exe 或者 vcredist_x86.exe）和 FastCGI 扩展（fcgisetup32.msi），以确保本机的 IIS 6 支持 FastCGI。

实例 009	安装 MySQL 光盘位置：光盘\MR\01\009	初级 趣味指数：★★★☆

实例说明

MySQL 是目前最为流行的开放源码的数据库，是完全网络化的跨平台的关系型数据库系统。它是由 MySQL AB 公司开发、发布并支持的。任何人都能从 Internet 上下载 MySQL 软件，并且不需支付任何费用。

在 MySQL 的官方网站（http://www.mysql.com）中可以下载到最新版本的 MySQL 服务器。

■ 关键技术

在 MySQL 的官方网站（http://www.mysql.com）中下载最新版本的 MySQL 服务器的具体步骤如下：

（1）打开 IE 浏览器，在地址栏中输入 http://www.mysql.com，进入 MySQL 的官方网站，如图 1.26 所示。

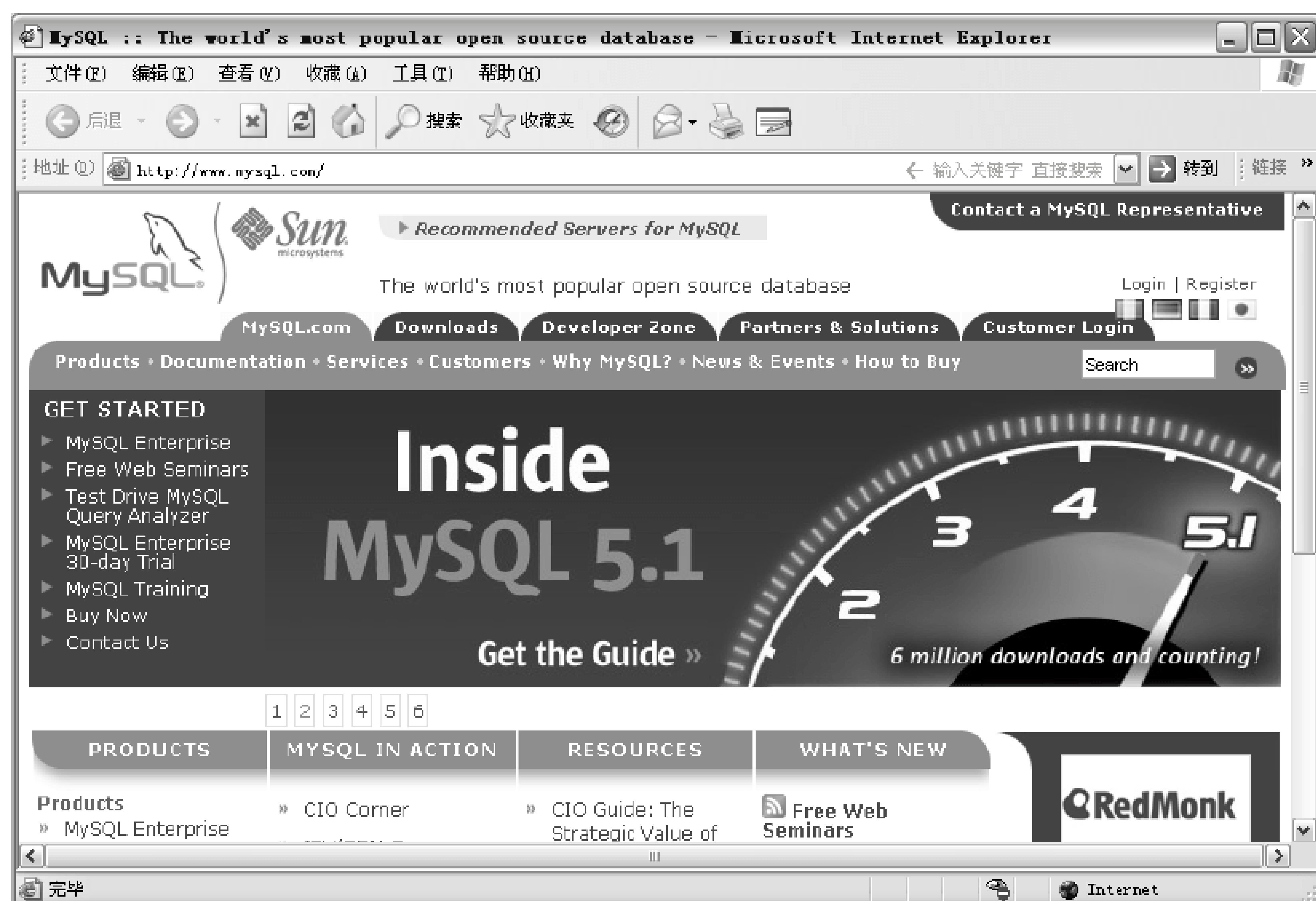


图 1.26　MySQL 的官方网站

（2）单击 Downloads 超链接，进入 MySQL 的 Downloads 页面，如图 1.27 所示。

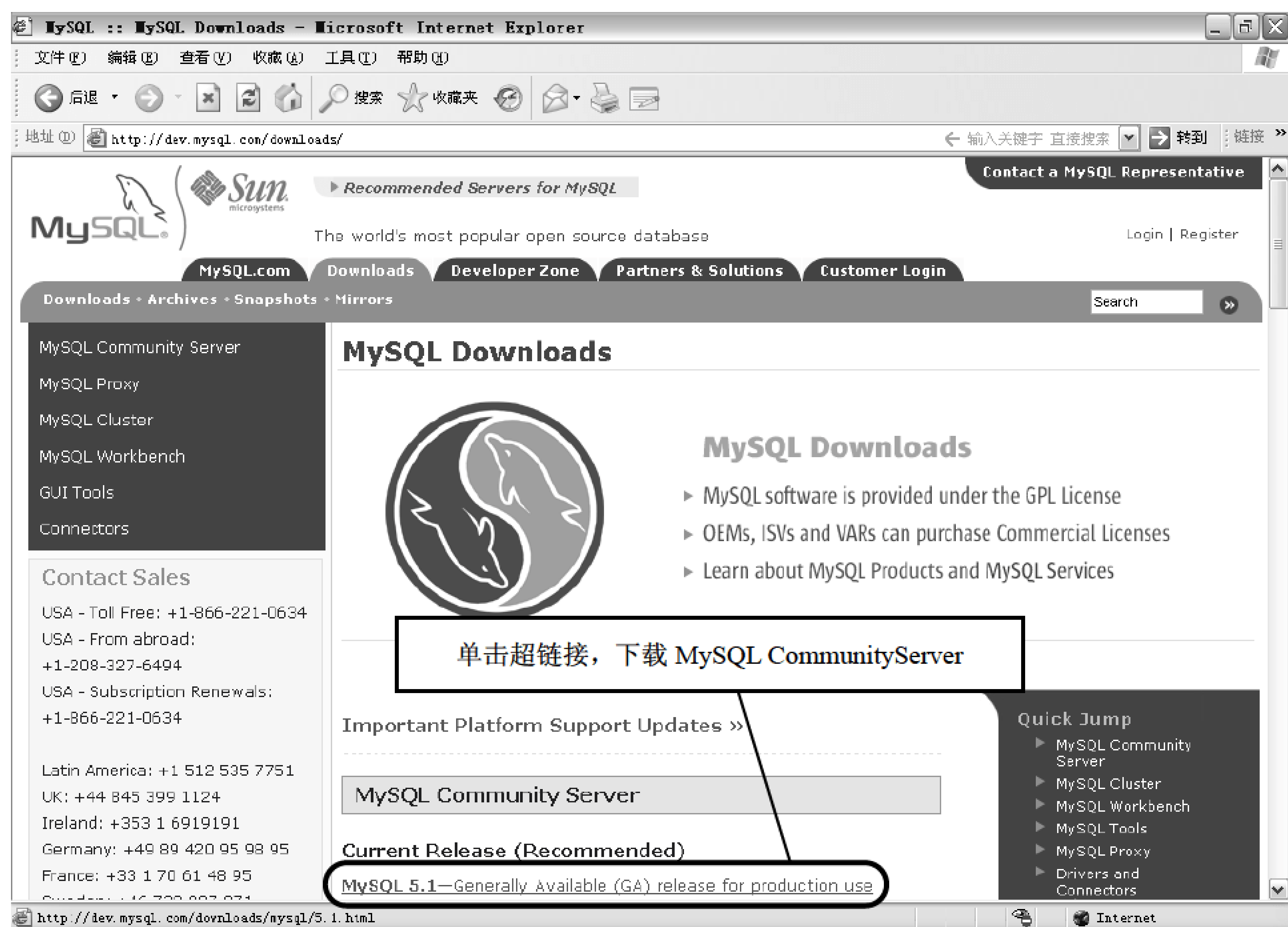


图 1.27　MySQL downloads 页面

（3）单击 MySQL 5.1—Generally Available（GA）release for production use 超链接，进入到 MySQL Connumity Server 下载页面，如图 1.28 所示。

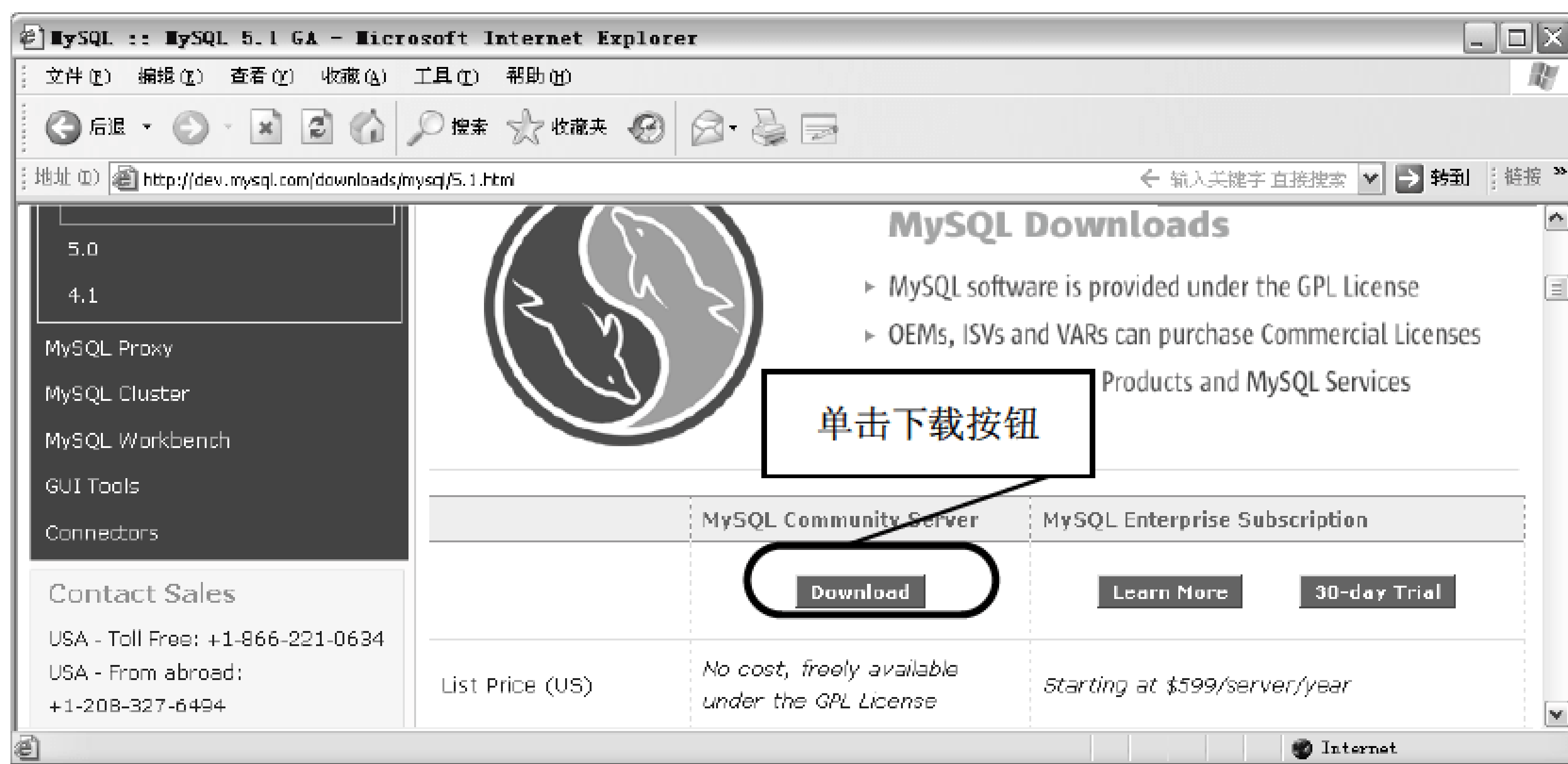


图 1.28　MySQL Connumity Server 下载页面

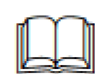
（4）单击 Download 按钮，进入到如图 1.29 所示的页面。在该页面中，可以选择安装的平台。



图 1.29　选择安装的平台

（5）这里选择 Windows 平台，单击 Windows 超链接，进入到如图 1.30 所示的页面。在该页面中，可以下载不同安装方式的 MySQL。

（6）单击 Windows Essentials(x86)后面的 Download 超链接，将弹出下载对话框，在该对话框中，单击“保存”按钮，下载 Windows 安装版的 MySQL。

 **说明：**（1）如果在该页面中，没有出现 Download 超链接，可以单击 Pick a mirror 超链接，选择一个镜像进行下载。

（2）本书在讲解 MySQL 的下载和安装时使用的是 mysql-essential-5.1.39 版本，当您看到本书时，这个版本可能不是最新的版本了。不过 MySQL 的下载和安装步骤基本没有变化，您仍然可以使用本书中提供的步骤、方法进行下载和安装。

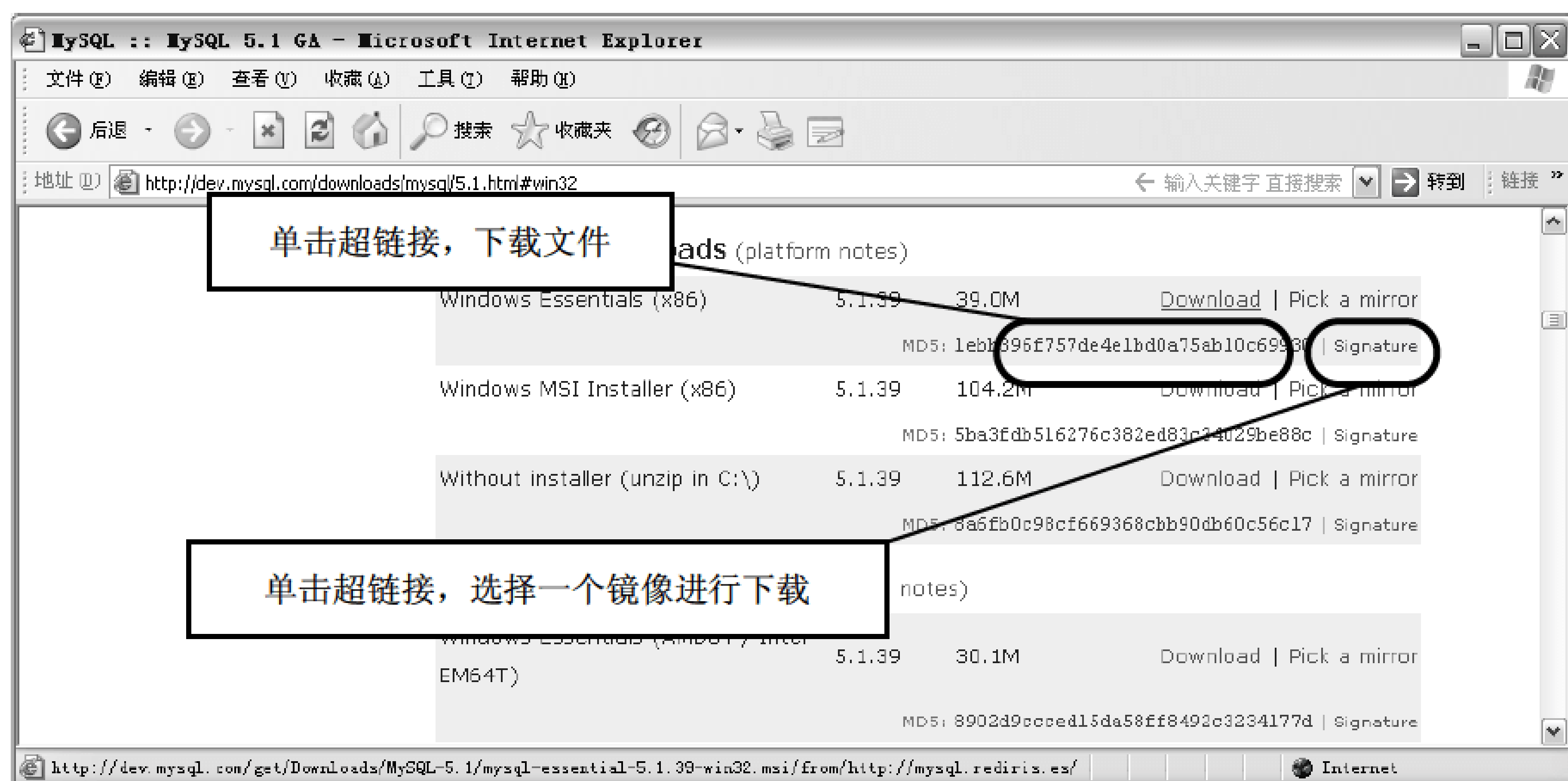


图 1.30　MySQL 的下载列表

设计过程

下面讲解 MySQL 的具体安装步骤。

（1）双击下载后的 mysql-essential-5.1.39-win32.msi 文件，打开安装向导对话框，单击 Next 按钮后，将打开 Setup Type（选择安装类型）对话框。在该对话框中，共包括 Typical（典型）、Complete（完全）和 Custom（自定义）3 种安装类型。这里选择典型安装，如图 1.31 所示。

（2）单击 Next 按钮，将打开准备安装对话框，在该对话框中将显示默认的安装路径，单击 Install 按钮，开始安装 MySQL。在安装的过程中，将显示 MySQL Enterprise 的宣传对话框，在该对话框中，单击 Next 按钮，进入到下一个宣传对话框，再次单击 Next 按钮，将打开如图 1.32 所示的 Wizard Completed（成功向导）对话框。在该对话框中，采用默认设置，单击 Finish 按钮，继续安装 MySQL 服务器。

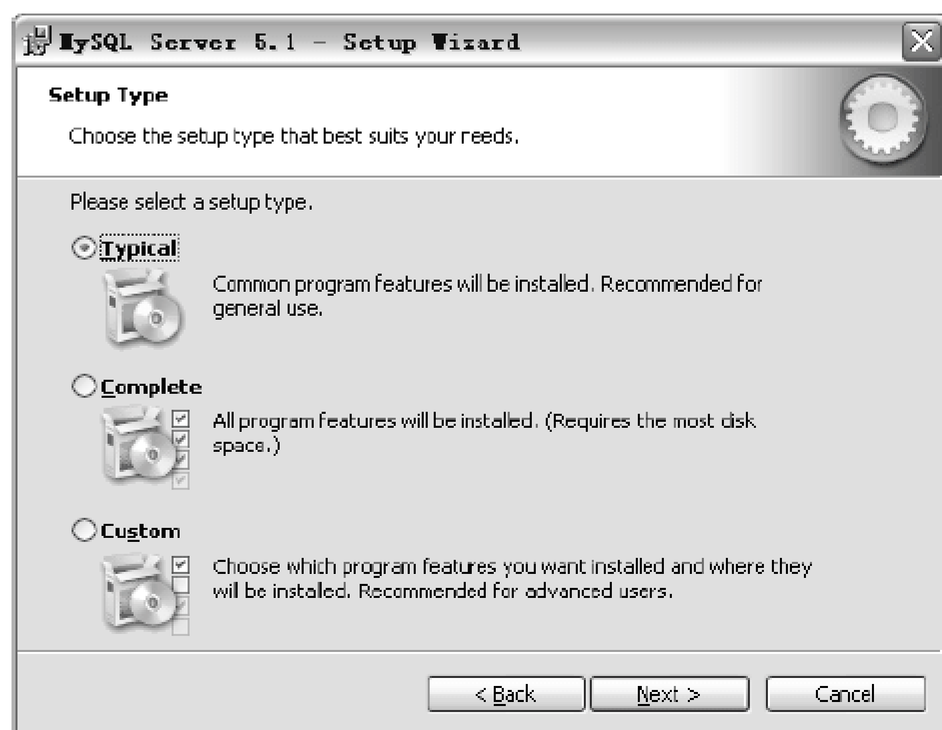


图 1.31　选择安装类型对话框

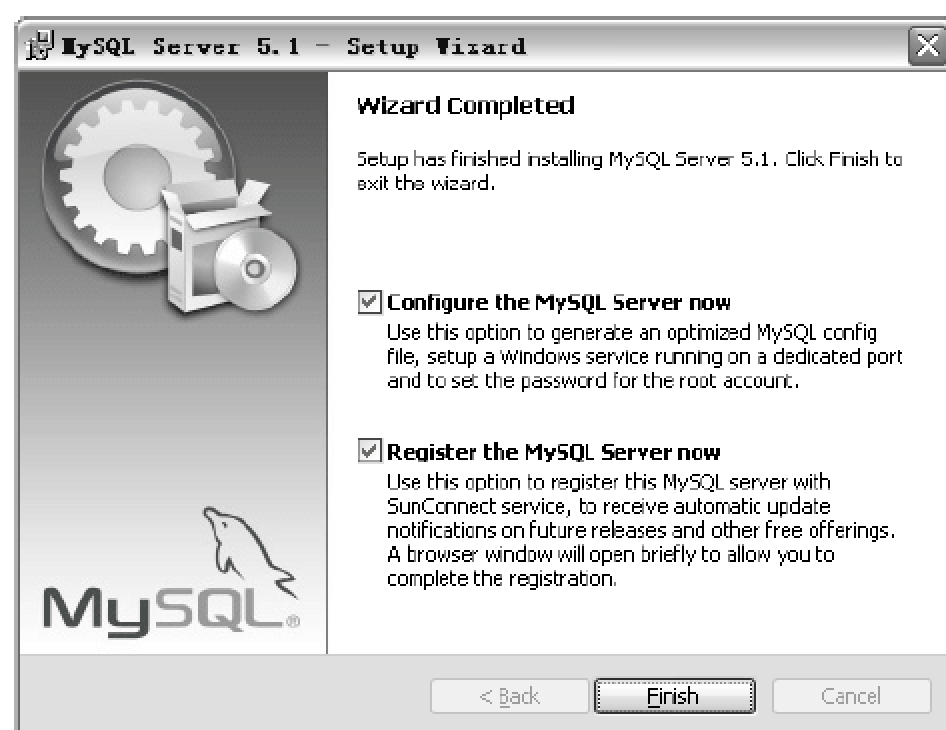


图 1.32　成功向导对话框

（3）这时将打开欢迎安装 MySQL 服务器的对话框，在该对话框中，单击 Next 按钮，将打开如图 1.33 所示的 MySQL 服务器配置类型对话框。在该对话框中，可以选择详细配置或者标准配置，这里选择详细配置。

（4）单击 Next 按钮，将打开如图 1.34 所示的选择服务器类型对话框。在该对话框中将提供开发者类型、服务器类型和致力于 MySQL 服务类型。这里选择默认的开发者类型。

（5）单击 Next 按钮，将打开如图 1.35 所示的选择数据库用法对话框，在该对话框中提供 3 个选项，分别是多功能型（Multifunctional Database）、仅提供事务型（Transactional Database Only）和仅不提供事务型（Non-Transactional Database Only）。这里选择多功能型。

（6）单击 Next 按钮，将打开如图 1.36 所示的设置服务器安装路径的对话框，这里采用默认设置。

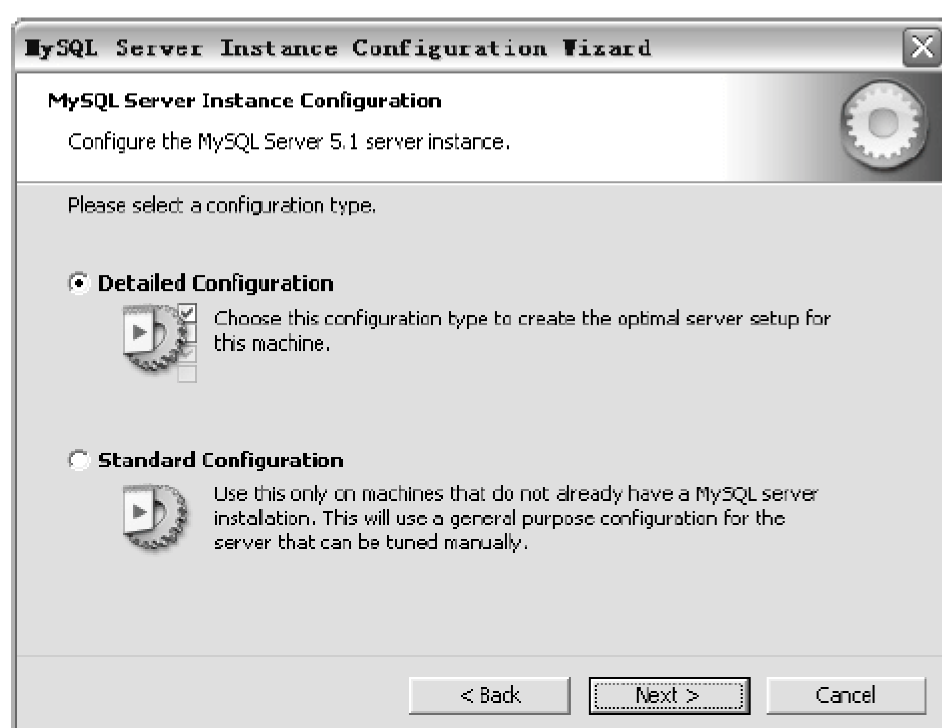


图 1.33　MySQL 服务器配置类型对话框



图 1.34　选择服务器类型对话框

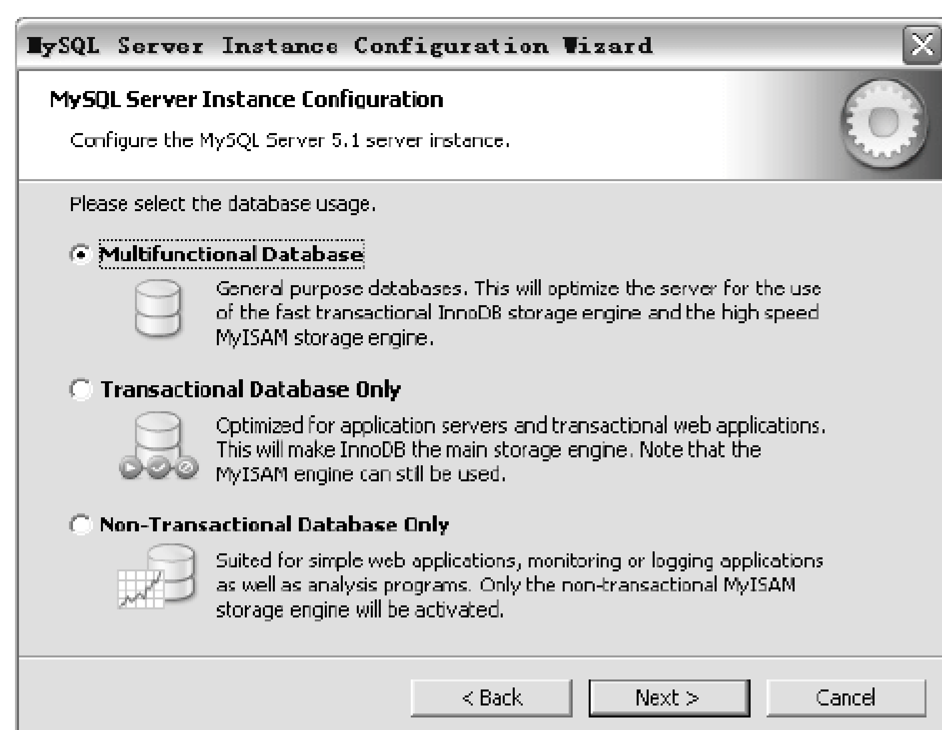


图 1.35　选择数据库用法对话框

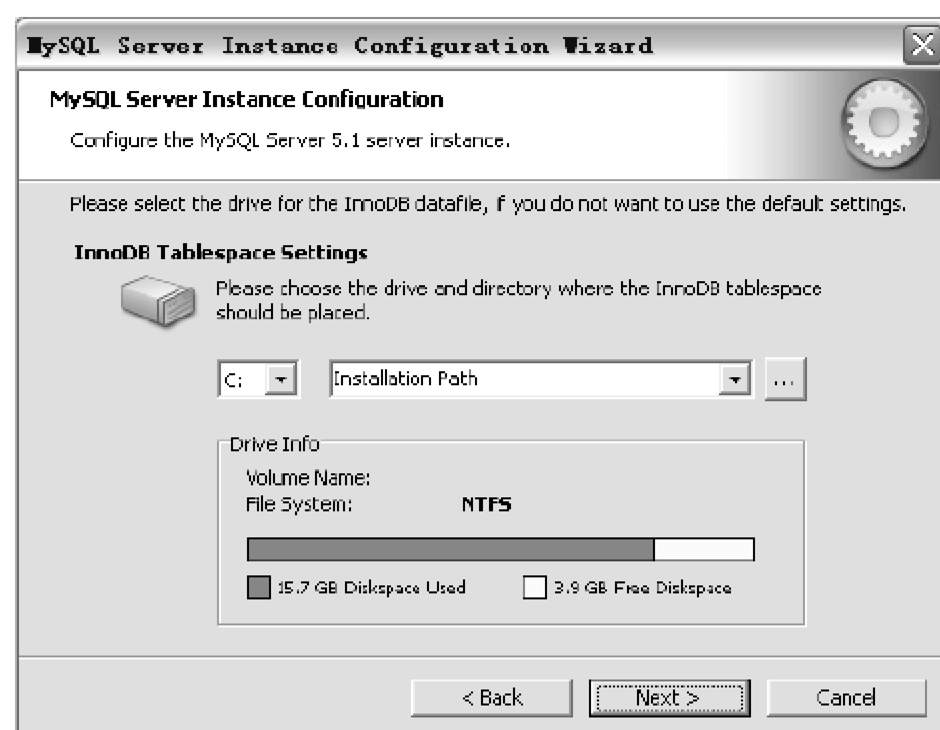


图 1.36　设置服务器安装路径对话框

（7）单击 Next 按钮，将打开如图 1.37 所示的设置提示连接的数量的对话框，这里采用默认设置。

（8）单击 Next 按钮，将打开如图 1.38 所示的设置网络选项对话框，在该对话框中采用默认设置。

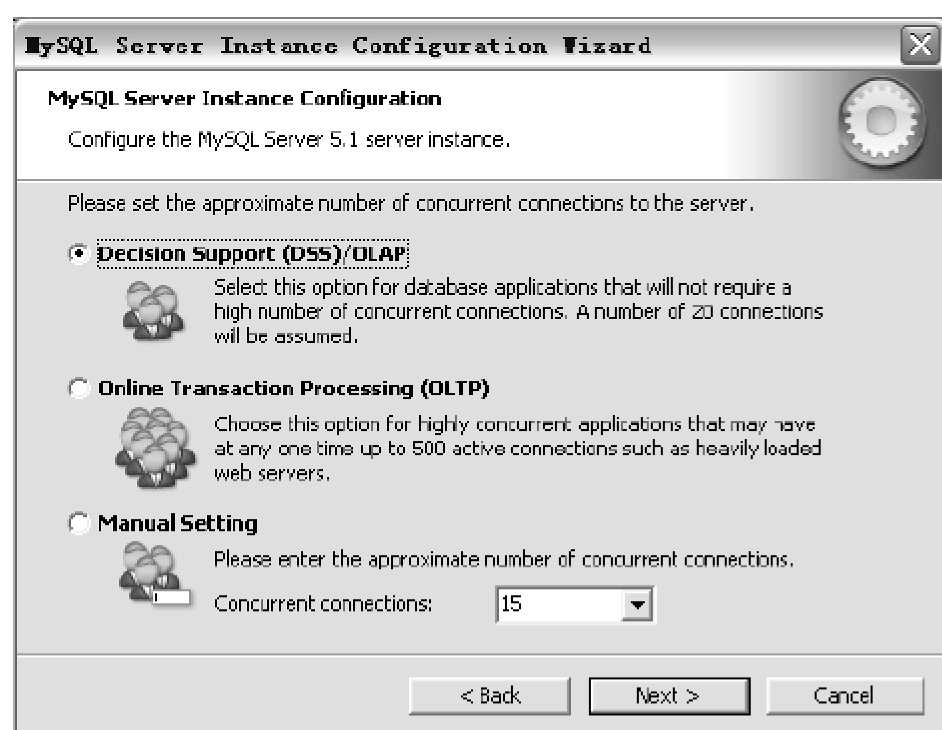


图 1.37　设置提示连接的数量的对话框

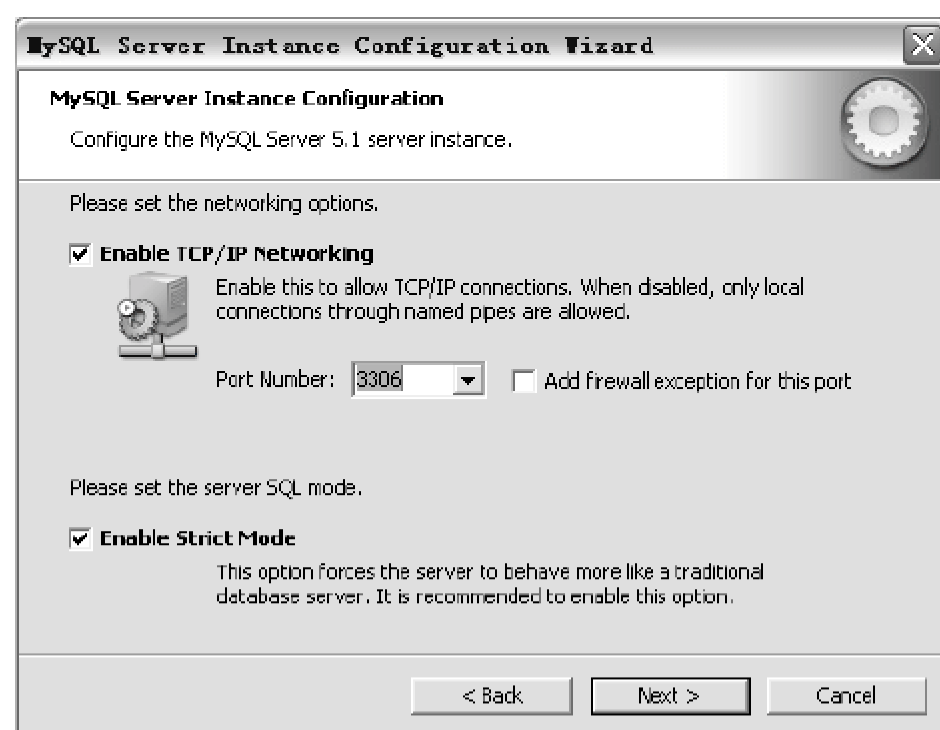


图 1.38　设置网络选项对话框

✍ **技巧：** MySQL 使用的默认端口是 3306，在安装时，可以修改为其他的，例如 3307。但是一般情况下，不要修改默认的端口号，除非 3306 端口已经被占用。

（9）单击 Next 按钮，将打开设置默认的字符集的对话框，在该对话框中选中 Manual Selected Default

Character Set/Collation 单选按钮，并在 Character Set（字符集）下拉列表框中选择 gbk（简体中文），如图 1.39 所示。

（10）单击 Next 按钮，将打开如图 1.40 所示的设置 Windows 选项的对话框，这里采用默认设置。



图 1.39　设置默认的字符集对话框

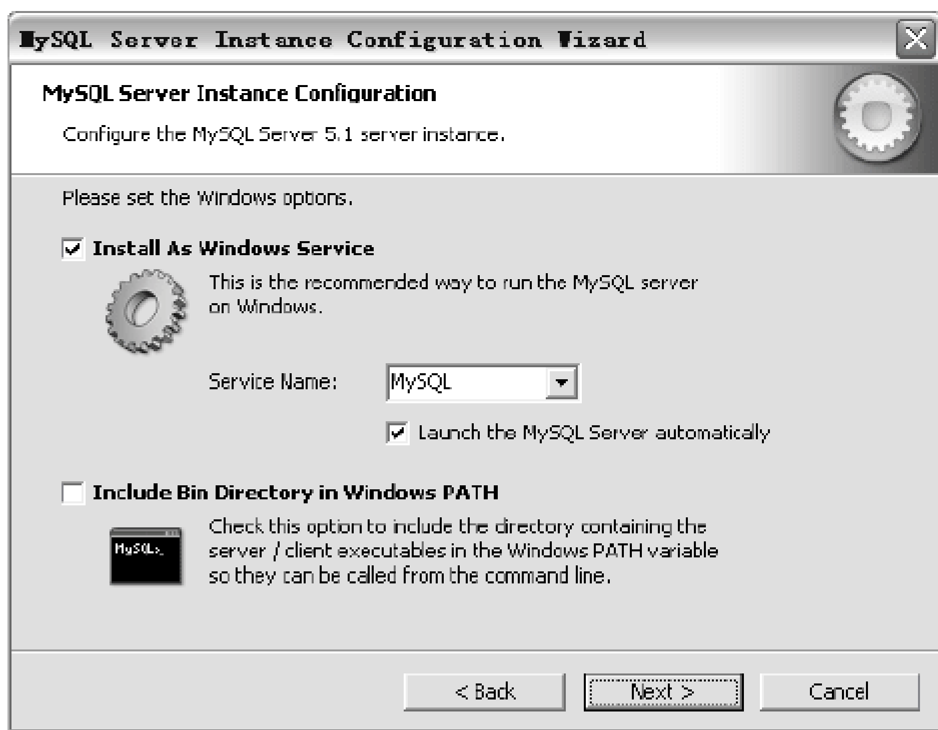


图 1.40　设置 Windows 选项对话框

（11）单击 Next 按钮，将打开安全设置选项对话框，该对话框用于为默认用户 root 设置密码，这里设置为 111，如图 1.41 所示。

（12）单击 Next 按钮，将打开准备执行对话框，单击 Execute 按钮，完成服务器配置，配置完成后，将显示如图 1.42 所示的对话框。单击 Finish 按钮，完成 MySQL 的安装。



图 1.41　安全设置选项对话框

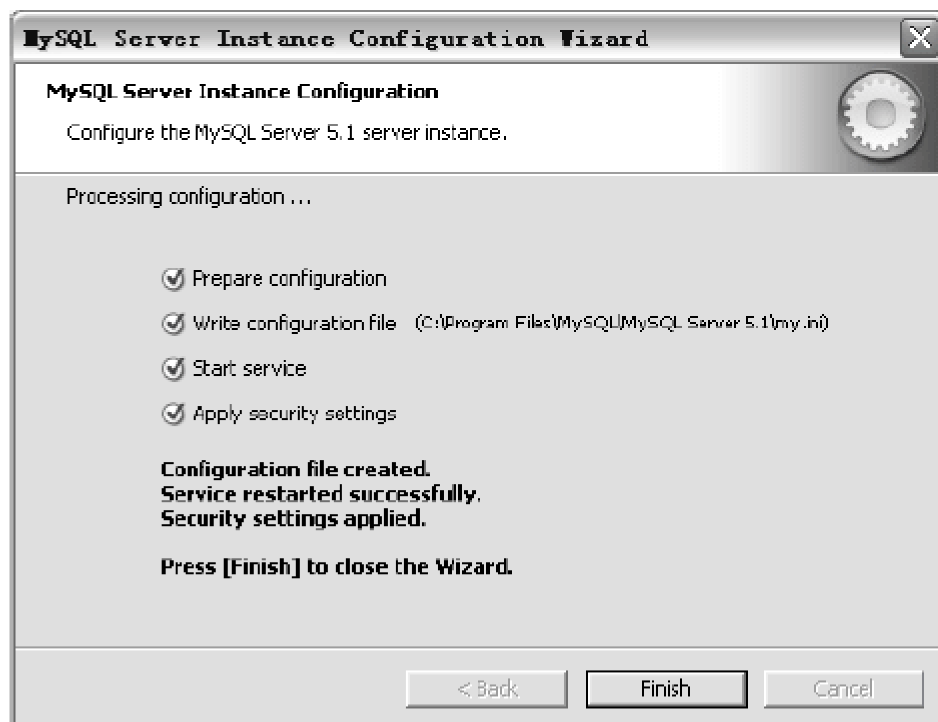


图 1.42　完成服务器配置

说明： MySQL 安装完成后，将在“开始”/“所有程序”菜单下，添加一个 MySQL 项，在该项中包括一个 MySQL Server 5.1 子项，在该子项下，又包括 MySQL Command Line Client、MySQL Server Instance Config Wizard 和 SunInventory Registration 3 个子项。其中 MySQL Command Line Client 为 MySQL 的客户端命令行，通过该项，可以查看数据表结构及执行 SQL 语句；MySQL Server Instance Config Wizard 项可用于修改 MySQL 的配置，例如修改字符集和 root 用户的密码等。

到此，MySQL 安装成功，如果要查看 MySQL 的安装配置信息，可以通过 MySQL 安装目录下的 my.ini 文件来完成。

在 my.ini 文件中，可以查看到 MySQL 服务器的端口号、MySQL 在本机的安装位置、MySQL 数据库文件存储的位置，以及 MySQL 数据库的编码等配置信息，如图 1.43 所示。

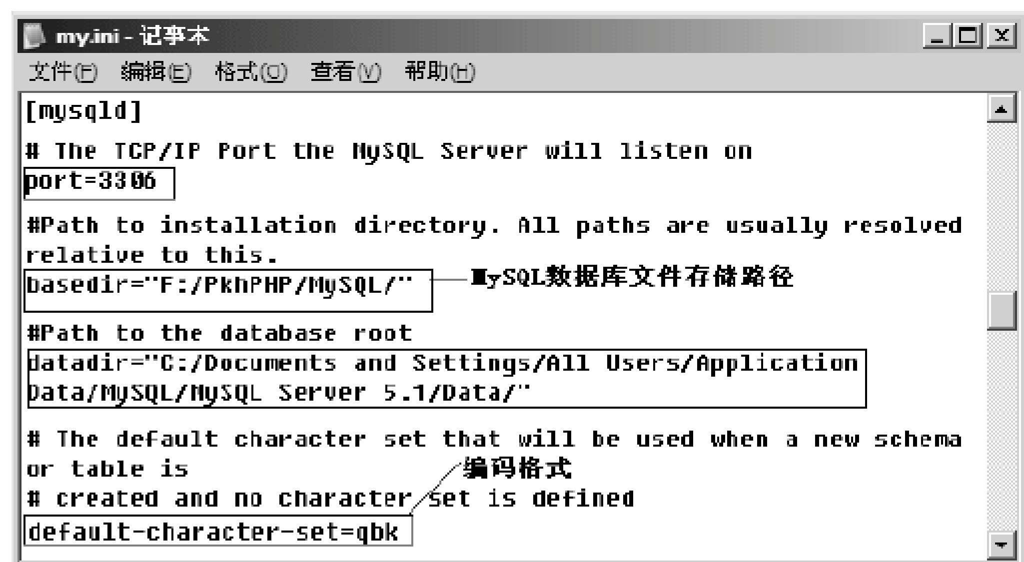


图 1.43　my.ini 文件的配置信息

秘笈心法

心法领悟 009：设置 MySQL 用户的密码。

在安装 MySQL 数据库时，一定要牢记在第 11 步中设置的 root 用户的密码，它是我们在访问 MySQL 数据库时必须使用的。

实例 010	安装 IIS 光盘位置：光盘\MR\01\010	初级 趣味指数：★★★☆

实例说明

本书以 Windows 2003 中使用的 IIS 6.0 为例，讲解 IIS 的安装。至于在其他操作系统中的安装步骤是相同的，只是使用的版本有所不同。

关键技术

IIS 的安装非常简单，关键是如何配置 IIS 服务器。

（1）选择“开始”/“设置”/“控制面板”命令，在弹出的窗口中双击“管理工具”图标，在弹出的窗口中单击“Internet 信息服务（IIS）管理器”图标，将弹出 IIS 管理器界面，如图 1.44 所示。

（2）图中的“默认网站”即为 IIS 服务器的文件目录。右键单击“默认网站”，在弹出的快捷菜单中选择“属性”命令，设置网站的属性，如图 1.45 所示。

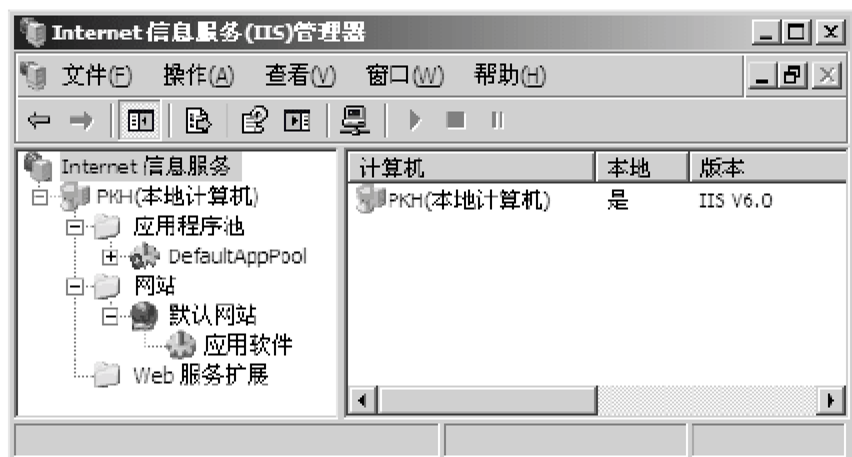


图 1.44　IIS 管理器界面

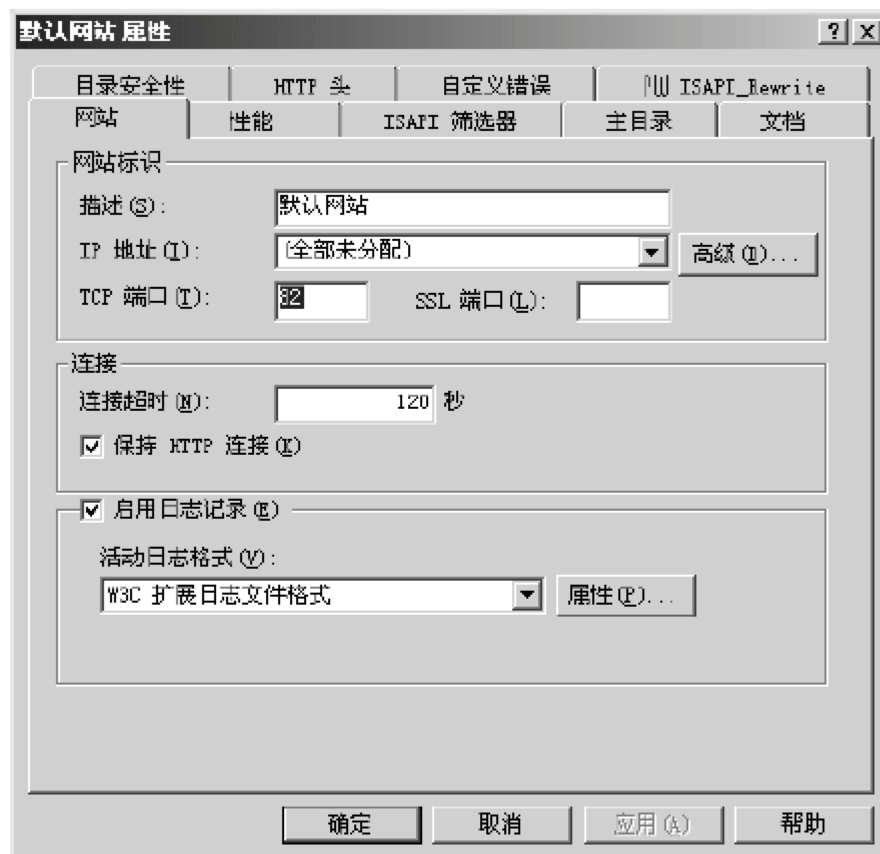


图 1.45　“网站”选项卡

（3）默认选中的是“网站”选项卡，在其中可以设置 TCP 端口号，这里将 IIS 端口号设置为“82”。设置完成后选择“主目录”选项卡，如图 1.46 所示。

（4）在“主目录”选项卡中进行 3 项设置：第 1 项是设置本地路径，用户可以自行定义，如 F:\PkhiisPHP\www；第 2 项是设置执行权限，这里设置为纯脚本；第 3 项是添加 PHP 的支持。单击“主目录”选项卡中的“配置”按钮，弹出如图 1.47 所示的“应用程序配置”对话框。

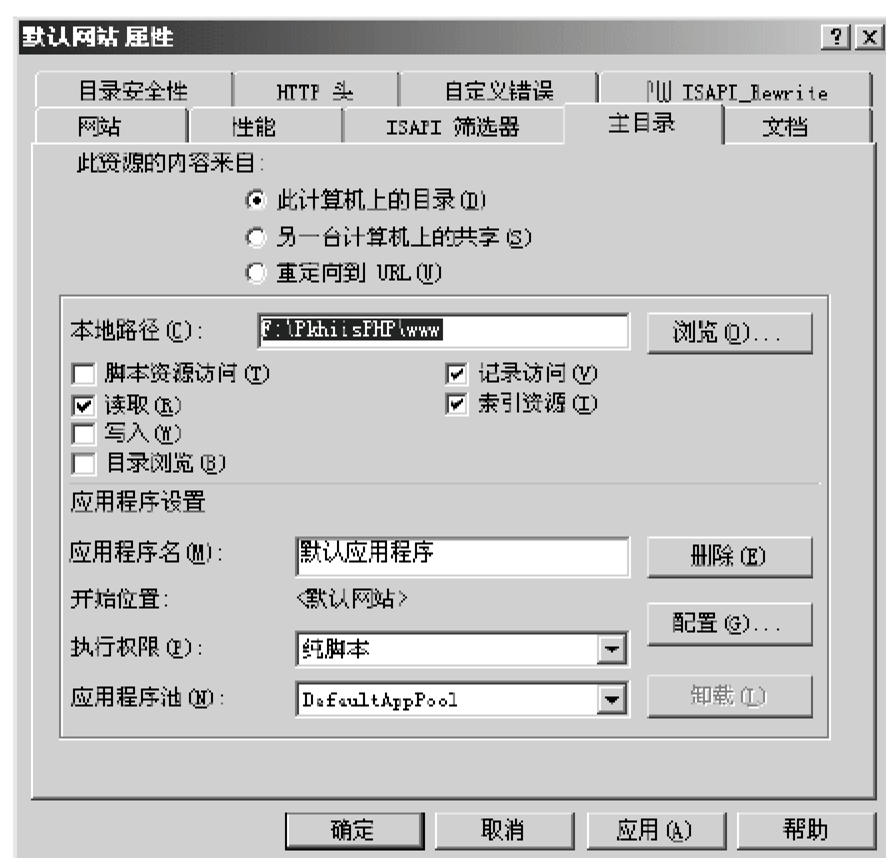


图 1.46 “主目录”选项卡



图 1.47 应用程序配置

（5）在图 1.47 所示对话框中，选择“映射”选项卡，完成添加、编辑应用程序扩展。单击“添加”按钮，在如图 1.48 所示的对话框中，首先指定可执行文件，这里指定的是存储在系统盘的 system32\inetsrv\文件夹下的 fcgiext.dll 文件，然后设置扩展名为“.php”，最后单击“确定”按钮，返回到“默认网站属性”对话框。

（6）选择“文档”选项卡，添加一个.php 的默认页。添加后的页面如图 1.49 所示。



图 1.48 添加 PHP 扩展

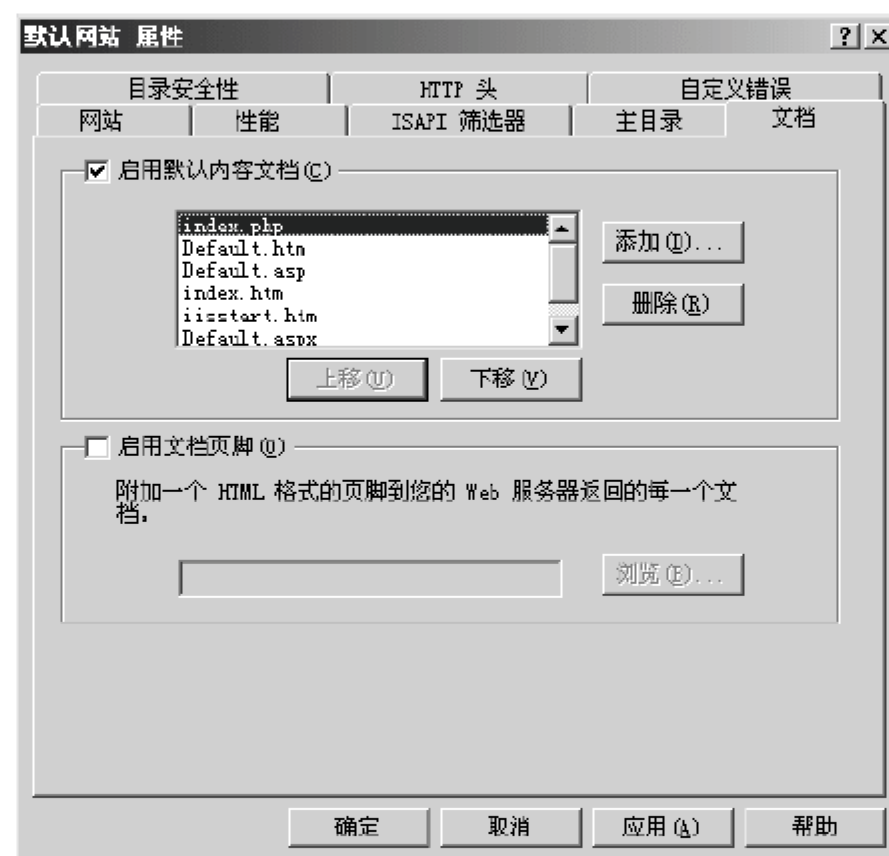


图 1.49 添加默认页

（7）关闭所有对话框，重启 IIS 服务器。

设计过程

（1）选择“开始”/“设置”/“控制面板”命令，在弹出的窗口中单击“添加/删除程序”按钮。

（2）单击左侧的“添加/删除 Windows 组件”按钮，将会弹出“Windows 组件向导”对话框。依次选中并双击“Windows 组件向导”对话框中的“应用程序服务器”选项、“应用程序服务器”窗口中的“Internet 信息

服务（IIS）”选项、“Internet 信息服务（IIS）”对话框中的“万维网服务”选项。选中“万维网服务”复选框。

（3）设置完毕后单击“确定”按钮。具体操作过程如图 1.50 所示。

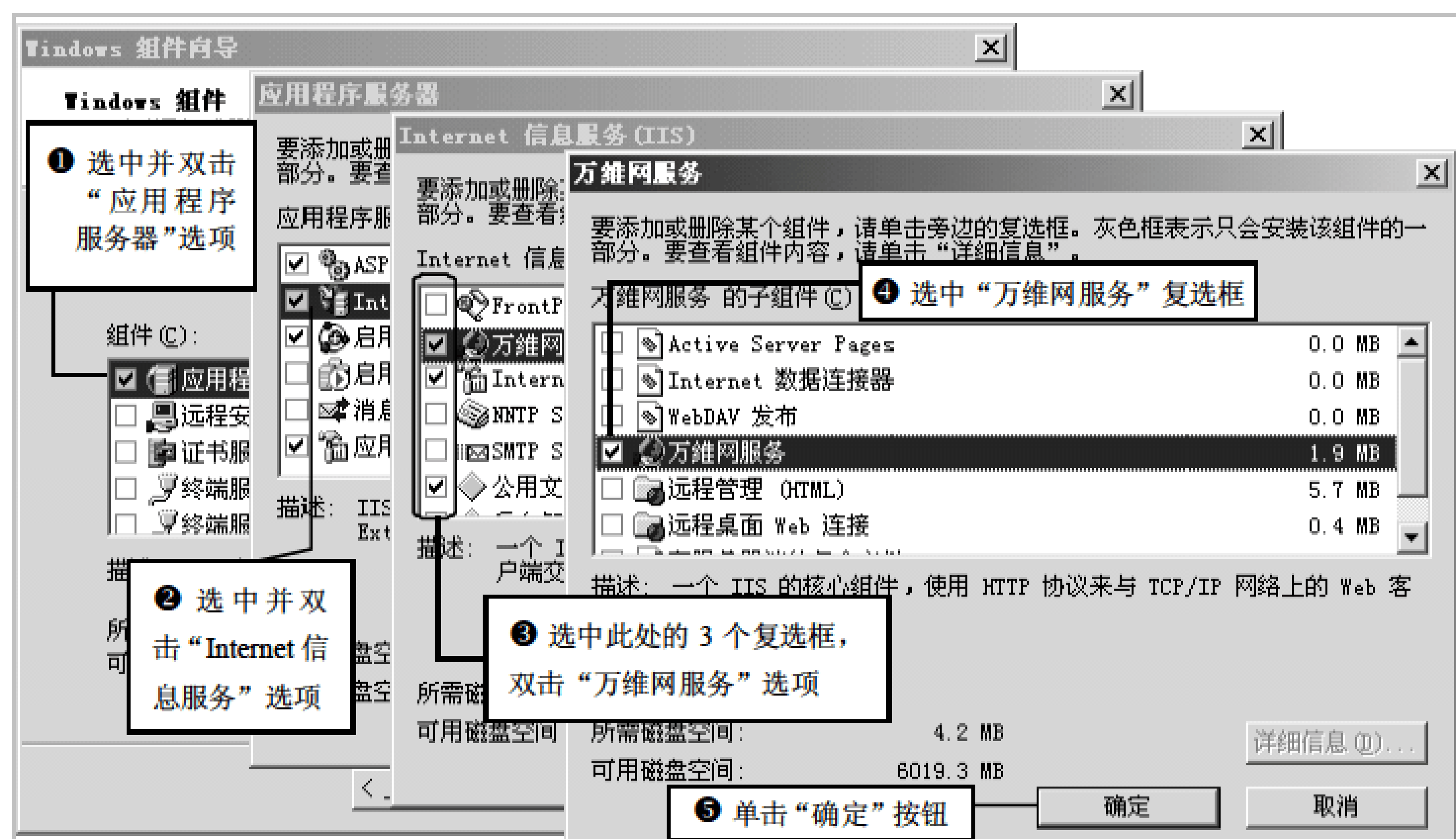


图 1.50　Internet 信息服务（IIS）

安装完成后，即可使用 IIS。

秘笈心法

心法领悟 010：测试 IIS 是否安装成功。

安装 IIS 后，即可在 IE 浏览器的地址栏中输入 http://localhost，测试 IIS 是否安装成功。

如果本机的 80 端口被 Apache 占用，那么就需要修改 IIS 的端口号。例如，修改为 82 端口，那么测试时应该在地址栏中输入 http://localhost:82/。

实例 011　**第 3 个 PHP 程序**　初级

光盘位置：光盘\MR\01\011　趣味指数：★★★☆

实例说明

本实例编写第 3 个 PHP 脚本，在 IIS 服务器下运行，测试通过 IIS+PHP+MySQL 搭建的 PHP 开发环境是否成功，运行结果如图 1.51 所示。

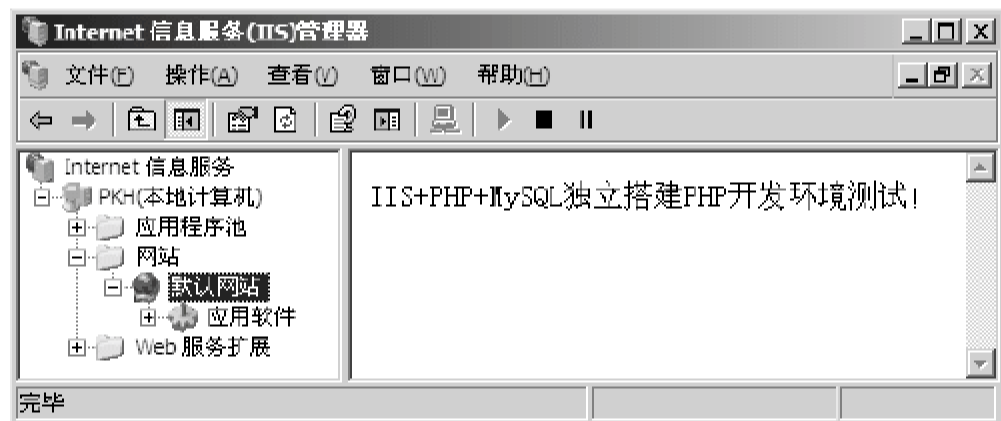


图 1.51　测试独立环境是否搭建成功

关键技术

确定 IIS 服务器设置的本地路径，这是在 IIS 的配置中完成的，必须将 PHP 脚本存储于 IIS 设置的本路径下才能够运行。

设计过程

（1）在 IIS 指定的本地路径下（F:\PkhiisPHP\www），新建一个文本文档，在该文档中输入如下内容：

```
<?php
echo "IIS+PHP+MySQL 独立搭建 PHP 开发环境测试！";
?>
```

（2）将该文档重命名为 index.php。

（3）选择“开始”/“设置”/“控制面板”命令，在弹出的窗口中双击“管理工具”图标，然后选择“Internet 信息服务（IIS）管理器”选项，依次展开“本地计算机”和“网站”节点，右键单击“默认网站”选项，选择“浏览”命令，若在“Internet 信息服务（IIS）管理器”窗口的右侧输出如图 1.51 所示的内容，说明独立环境搭建成功。

秘笈心法

心法领悟 011：IIS 中的 UrlRewrite 技术。

UrlRewrite 即地址变换，可以实现静态功能，方便搜索引擎收录，比如 http://***/news.php?id=1 可以变为 htt://***/news-1.html。不但可以实现开发标记的隐藏，而且避免每个页面都输出.php 后缀，进而提高网站的安全性。

IIS 默认是不支持此功能的，需要单独安装插件 ISAPI_Rewrite3_0048_Lite.msi。 安装完成后打开 IIS，在 WEB 属性的“ISAPI 筛选器”里添加位于安装目录的 Helicon\ISAPI_Rewrite3 文件夹里的 ISAPI_Rewrite.dll 即可，名称可以自定义。

1.4　LAMP——独立配置 PHP 开发环境

在 Linux 下搭建 PHP 环境比在 Windows 中要复杂一些，除了 Apache、PHP 等软件外，还要安装一些相关工具，并设置必要参数。这里给出在 Linux 下搭建 PHP 环境的必要步骤，如果用户在安装过程中遇到特殊的问题，还需要翻阅与 Linux 相关的资料、书籍或者手册。

安装之前要准备的安装包如下：

☑ apache_1.3.41.tar.gz

☑ php-5.0.0.tar.gz

☑ mysql-standard-5.0.0-alpha-pc-linux-i686.tar.gz

☑ libxml2-2.6.19.tar.gz

说明： 本书中在讲解 Linux 下搭建 PHP 开发环境时，使用的软件版本相对较低，如果读者想要下载这些软件，可以登录 http://221.8.65.77/mrbook/bookzy/linux.rar。

这里首先在 Linux 系统的/usr/local/文件夹下新建一个 work 文件夹，并将使用的软件全部存储在此文件夹下。

实例 012	安装 Apache 服务器 光盘位置：光盘\MR\01\012	中级 趣味指数：★★★★☆

实例说明

本实例讲解在 Linux 操作系统下安装 Apache 服务器。Apache 服务器安装成功后测试的运行结果如图 1.52 所示。



图 1.52　Apache 配置成功

关键技术

Linux 系统下配置 Apache 服务器使用的仍是 httpd.conf 文件，该文件位于/usr/local/apache/conf 中。

（1）在 httpd.conf 文件中，定位到如图 1.53 所示的位置，添加相应的内容，并设置 PHP 文件的后缀。

（2）同样在 httpd.conf 文件中，定位到如图 1.54 所示的位置，设置 PHP 文件的默认页。

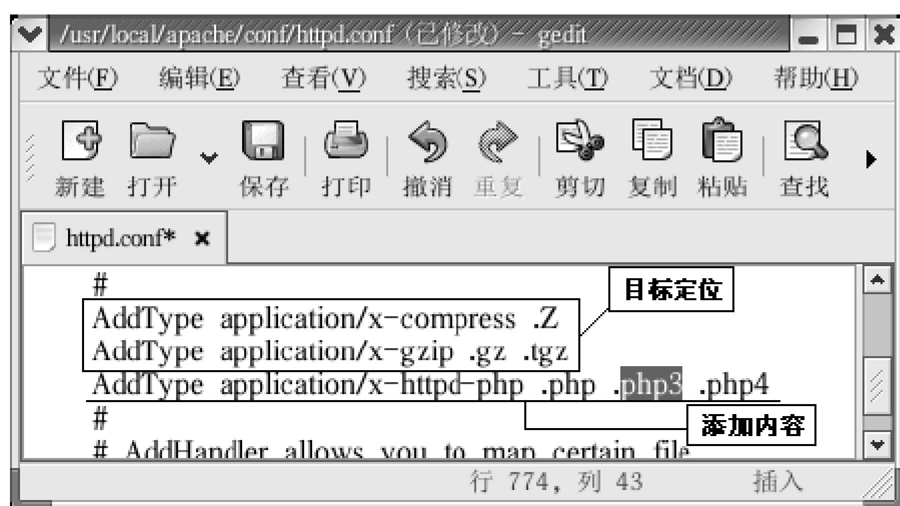


图 1.53　设置文件后缀



图 1.54　设置默认文件

设计过程

安装 Apache 服务器，首先需要打开 Linux 终端（Linux 下几乎所有的软件都需要在终端下安装）。选择 RedHat9 的“主菜单”/“系统工具”命令，在弹出的子菜单中选择“终端”命令。Apache 服务器的安装步骤如下：

（1）解压 apache_1.3.41.tar.gz 压缩包，将其存储到 url/local/apache 目录下，执行的命令如图 1.55 所示。

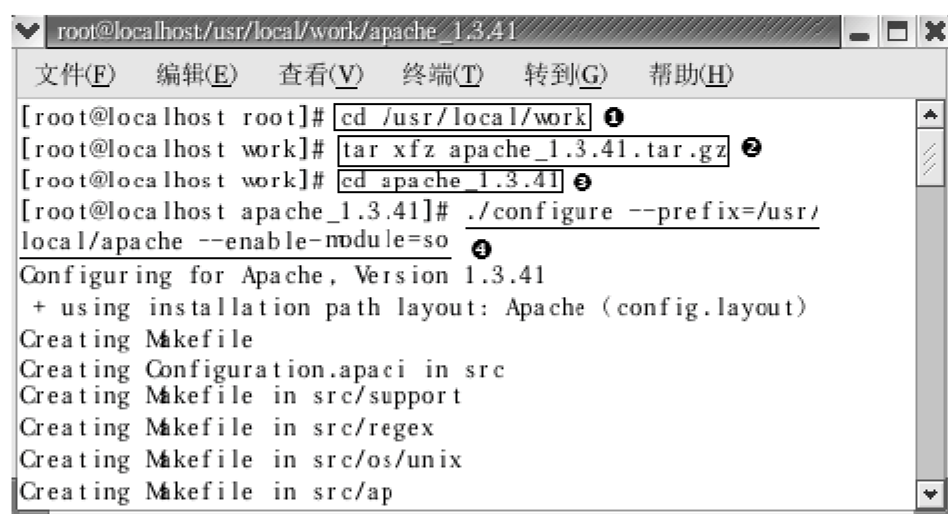
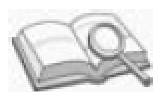


图 1.55　输入命令



命令导读

❶ 进入 Apache 安装文件的目录。

❷ 解压 Apache 服务器的安装包。

❸ 进入 apache_1.3.41 目录。

❹ 将 Apache 服务器安装到 usr/local/apache 目录下。

（2）执行 make 命令，编译 apache 文件。执行命令如图 1.56 所示。

（3）执行 make install 命令，开始安装 Apache 服务器。执行命令如图 1.57 所示。

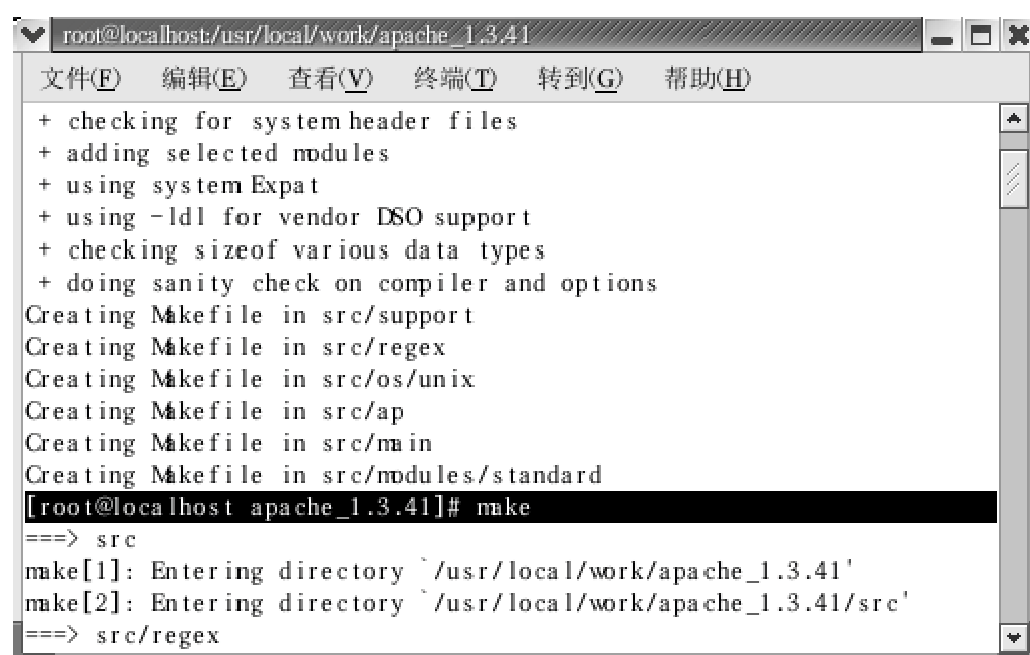


图 1.56　执行 make 命令编译文件



图 1.57　执行 make install 安装命令

（4）Apache 服务器安装成功后，将给出如图 1.58 所示的命令提示。

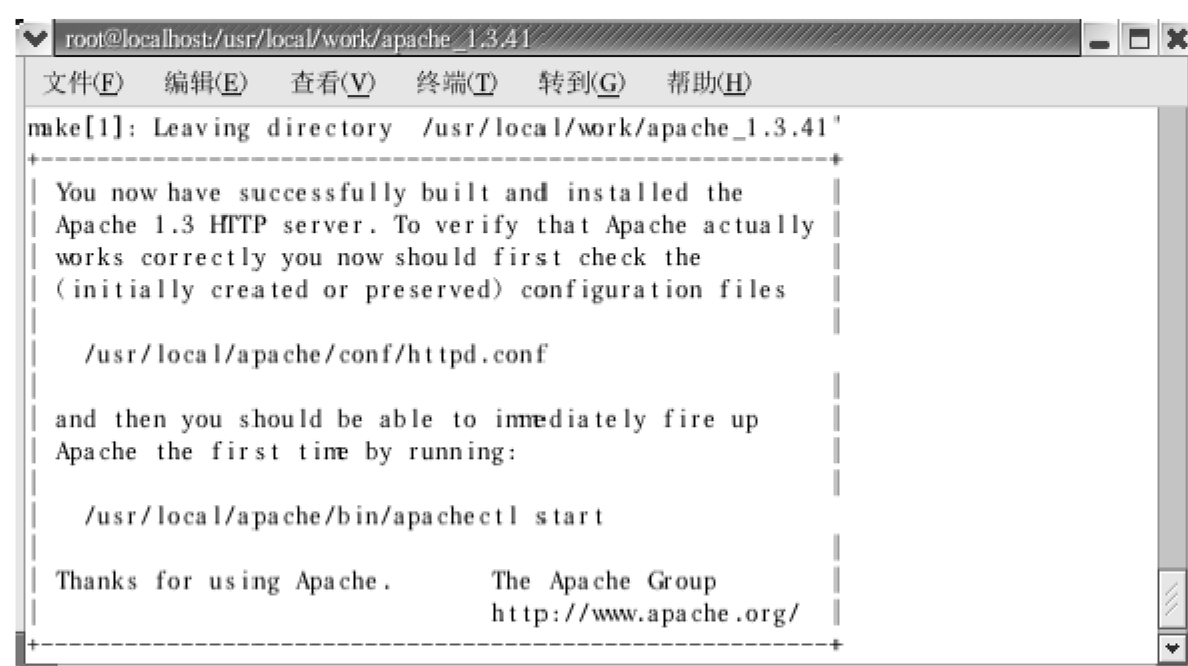


图 1.58　安装成功返回命令

秘笈心法

心法领悟 012：Linux 下启动 Apache。

安装成功后，就可以启动 Apache 服务器了，启动 Apache 服务器使用的命令如图 1.59 所示。

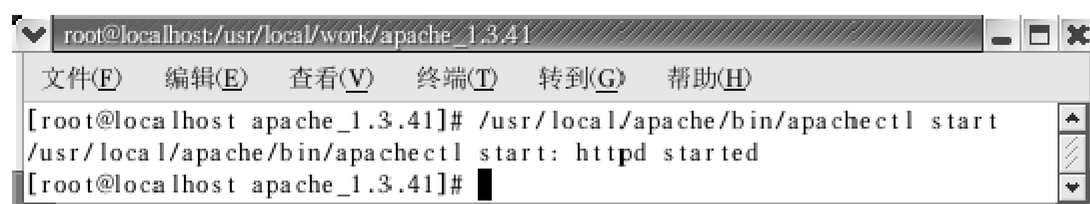


图 1.59　执行启动 Apache 服务器的命令

实例 013　安装 MySQL 数据库服务器

光盘位置：光盘\MR\01\013

中级

趣味指数：★★★★☆

实例说明

本实例讲解在 Linux 下安装 MySQL 数据库，使用的版本是 mysql-standard-5.0.0- alpha- pc- linux-i686.tar.gz。

关键技术

（1）在 Linux 操作系统中，备份和还原 MySQL 使用的命令如下：

备份：

```
mysqldump –uroot –proot –R 数据库名称>C:/data.sql
```

还原：

```
mysql –uroot –proot –D 数据库名称< C:/data.sql
```

（2）在 Linux 操作系统中，设置和修改 root 用户密码的命令如下：

设置密码：

```
mysqladmin –uroot password root
```

修改密码：

```
mysqladmin –uroot –p 旧密码 password 新密码
```

设计过程

下面讲解其具体的安装步骤。

（1）安装 MySQL 数据库同样要在 Linux 终端的命令中进行。首先创建 MySQL 账号，并将新建账号加入到组群，具体使用的操作命令如图 1.60 所示。

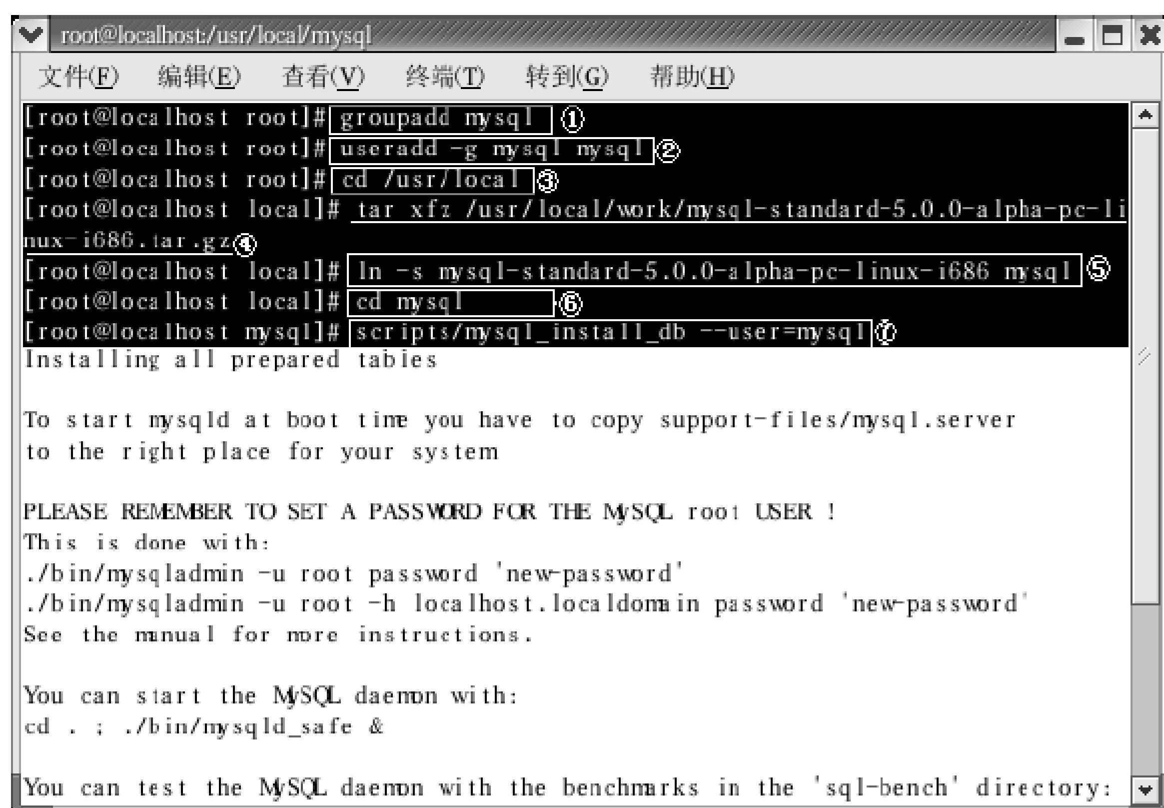


图 1.60　MySQL 的安装命令



代码导读

❶ 创建 MySQL 账号。
❷ 加入组群。
❸ 进入 MySQL 安装文件的目录。
❹ 解压 MySQL 服务器软件。
❺ 以链接的方式建立/usr/local/mysql 目录。
❻ 进入 MySQL 目录。
❼ 在/usr/local/mysql/data 中建立 MySQL 数据库。

（2）在完成数据库的安装之后，修改文件的权限。具体操作命令如图 1.61 所示。

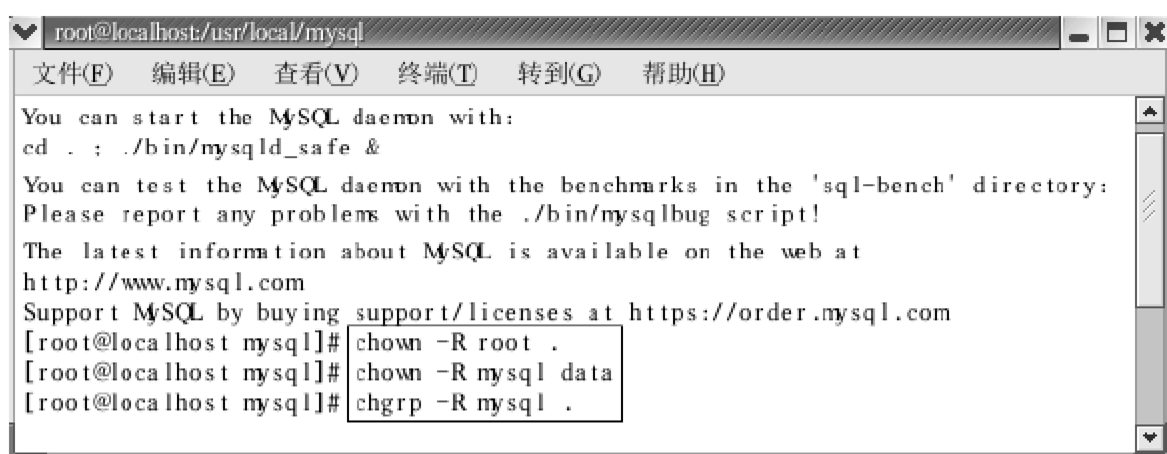


图 1.61　修改文件权限的命令

（3）完成权限修改后，看一下启动 MySQL 服务器的命令。具体命令如图 1.62 所示。

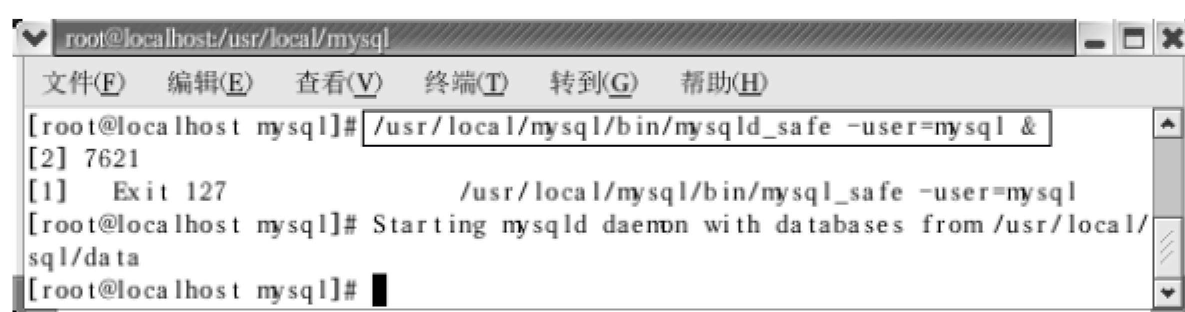


图 1.62　启动 MySQL 的命令

秘笈心法

心法领悟 013：Linux 下修改 MySQL 数据库用户的密码。

可以通过命令修改 MySQL 数据库用户的密码。在如图 1.63 所示的命令中，首先以无密码的方式登录 MySQL 数据库，然后通过命令设置 MySQL 数据库用户的密码，最后通过新设置的密码重新登录 MySQL 数据库。

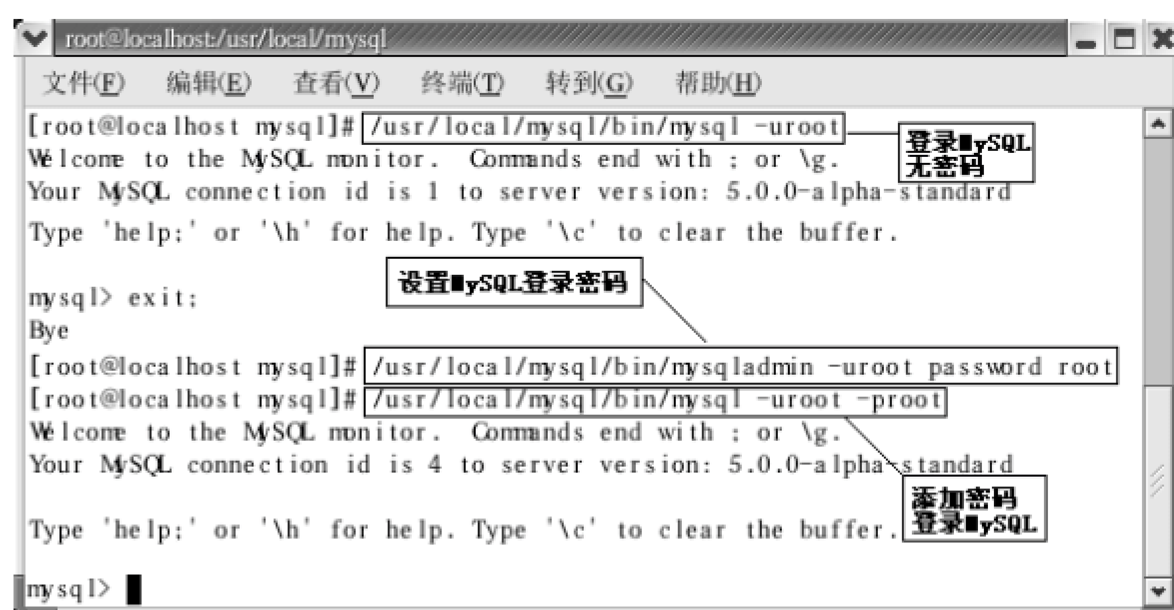


图 1.63　登录 MySQL 和设置 MySQL 登录密码

实例 014	安装 PHP.50 光盘位置：光盘\MR\01\014	中级 趣味指数：★★★★☆

实例说明

本实例讲解在 Linux 操作系统下安装 PHP 5.0。安装之前，首先需要查看 libxml 的版本号，如果小于 2.5.10，则需要先安装 libxml 的高版本，否则，可以直接安装 PHP 5.0。

关键技术

这里以 libxml2-2.6.19.tar.gz 为例，讲解 libxml 的安装步骤。

（1）将 libxml2-2.6.19.tar.gz 进行解压，并将其安装到指定的文件夹下，其操作命令如图 1.64 所示。

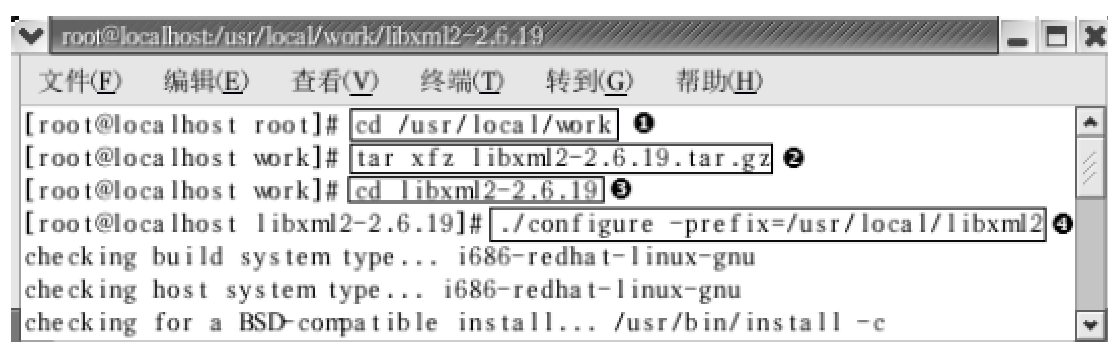


图 1.64　解压 libxml


代码导读

❶ 进入 usr/local/work 文件夹（libxml2-2.6.19 压缩包存储的位置）。

❷ 解压 libxml2-2.6.19.tar.gz。
❸ 进入 libxml2-2.6.19 文件夹。
❹ 将 libxml2-2.6.19 安装到/usr/local/libxml2。

（2）完成 libxml 的解压后，开始编译文件，使用的命令如图 1.65 所示。
（3）文件编译完成后，就可以执行安装操作了，使用的命令如图 1.66 所示。



图 1.65　编译文件

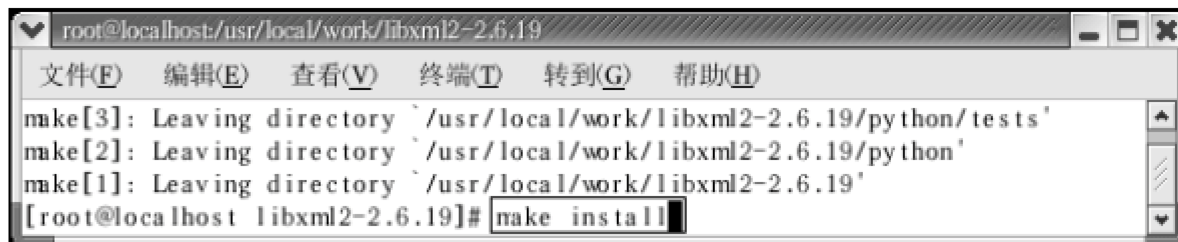


图 1.66　执行安装

设计过程

本书使用的版本是 php-5.0.0.tar.gz。下面来看具体的安装步骤。

（1）将 php-5.0.0.tar.gz 解压到指定的文件夹下，通过命令进行配置，使用的操作命令如图 1.67 所示。

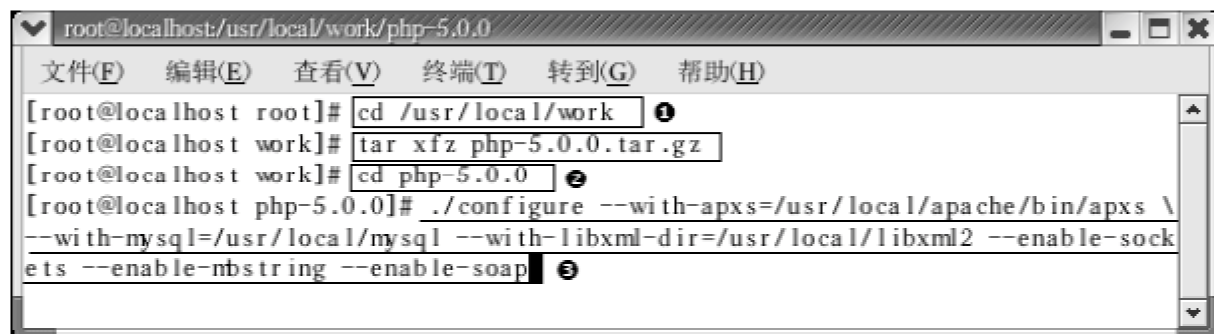


图 1.67　安装 PHP



代码导读

❶ 进入 usr/local/work 文件夹（php-5.0.0.tar.gz 压缩包存储的位置）。
❷ 解压 php-5.0.0.tar.gz。
❸ 配置文件。

（2）完成解压和配置操作后，对文件进行编译，使用的命令如下：

```
make
```

（3）编译完成后，执行安装操作，使用的命令如下：

```
make install
```

（4）最后，在命令中将出现如图 1.68 所示的内容，说明 PHP 安装成功。

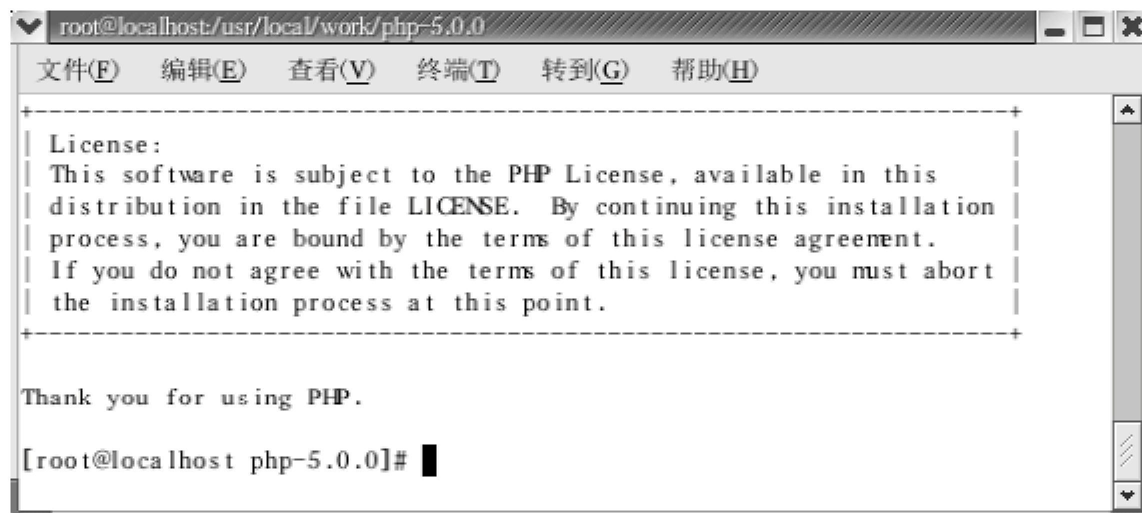


图 1.68　PHP 安装成功

秘笈心法

心法领悟 014：在 Linux 下创建 php.ini。

在 Linux 操作系统中，PHP 的配置文件仍然是 php.ini，其创建方法如下：

在 PHP 文件夹下找到 php.ini-dist 或者 php.ini-recommended，将其复制到/usr/local/lib 目录下，并重命名为 php.ini。

```
cp php.ini-dist /usr/local/lib/php.ini
```

实例 015	第 4 个 PHP 程序 光盘位置：光盘\MR\01\015	中级 趣味指数：★★★★☆

实例说明

在 Linux 系统下搭建完成 PHP 开发环境后，编写一个小实例，测试一下环境是否搭建成功。在 Mozilla 浏览器的地址栏中输入 http://localhost/test.php，运行结果如图 1.69 所示。

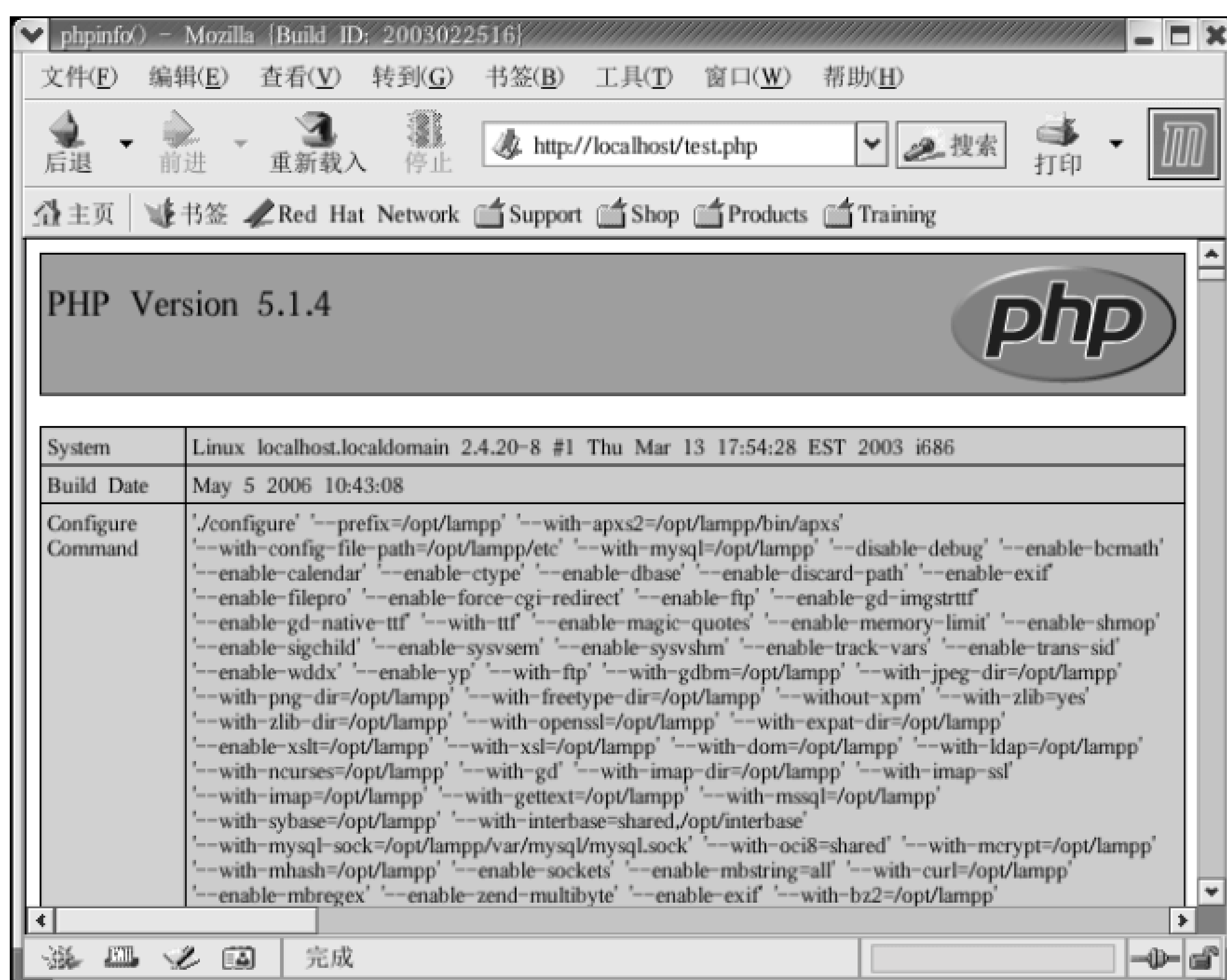


图 1.69　测试运行结果

关键技术

在 Linux 操作系统下，通过 gedit 工具编写 PHP 脚本。

设计过程

通过 gedit 工具编写一个 PHP 脚本 test.php，并存储于 Apache 的 htdocs 文件夹下。test.php 的代码如图 1.70 所示。



图 1.70　test.php 文件的内容

秘笈心法

心法领悟 015：Linux 下安装 XAMPP 集成环境。

本实例介绍的是在 Linux 操作系统下独立搭建 PHP 的开发环境，读者也可以使用 XAMPP 的 xampp-linux-1.7.3a.tar.gz 版本直接在 Linux 操作系统下搭建集成的开发环境。

1.5　XAMPP——Linux 版 PHP 集成化安装包

前面已经介绍过如何在 Windows 操作系统下应用 XAMPP 集成化安装包配置 PHP 的开发环境，这里将介绍如何在 Linux 操作系统下通过 XAMPP 配置 PHP 的开发环境以及其基本应用。

实例 016	XAMPP——Linux 下 PHP 开发环境的集成化 光盘位置：光盘\MR\01\016	中级 趣味指数：★★★★☆

实例说明

在 Linux 操作系统下，也可以使用集成的软件来配置 PHP 开发环境，它就是 XAMPP 的 Linux 版。配置成功后，在 Mozilla 浏览器的地址栏中输入 http://127.0.0.1/index.php，运行结果如图 1.71 所示。

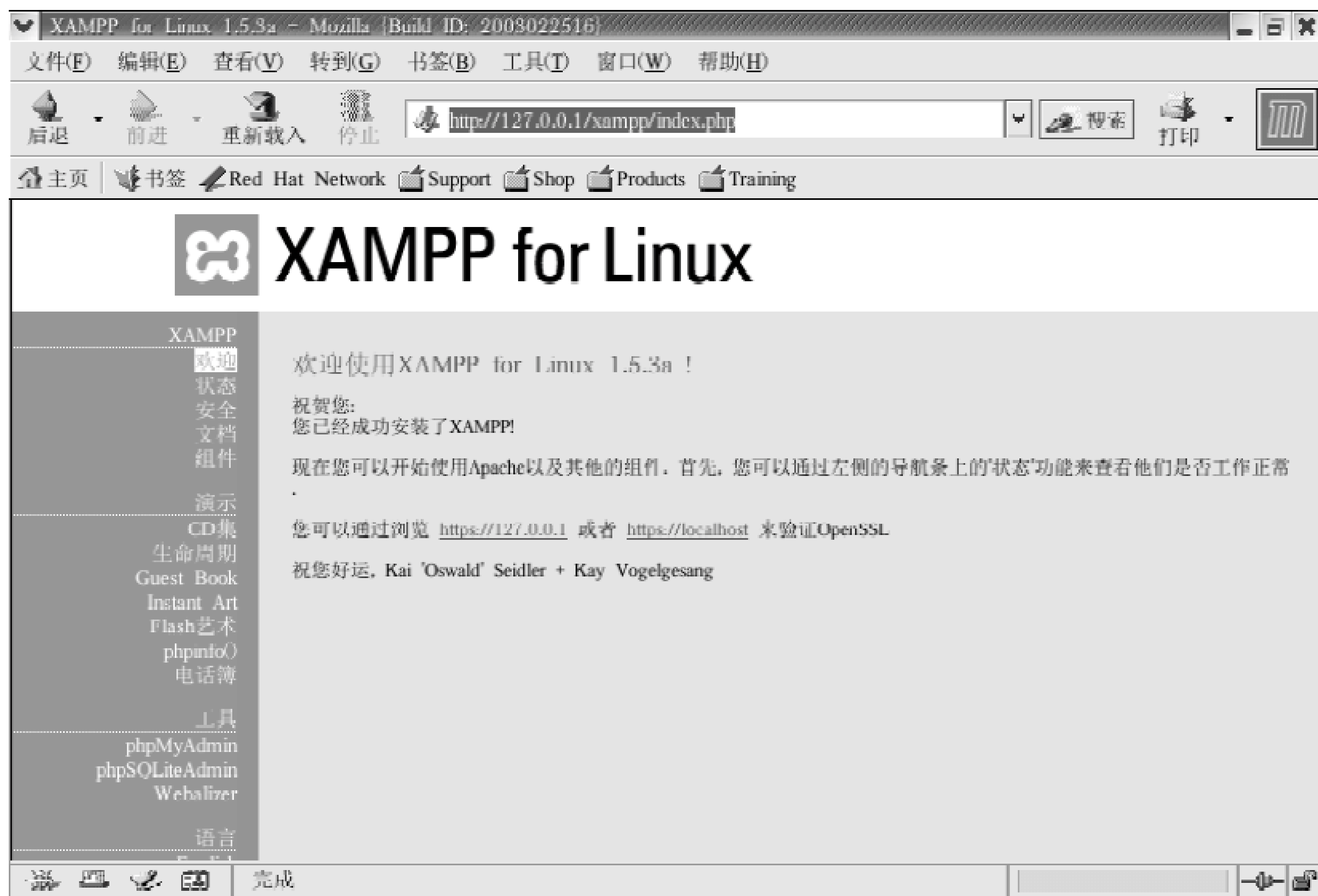


图 1.71　Linux 版的 XAMPP

关键技术

在 Linux 操作系统下安装 XAMPP，必须在 Linux “主菜单”/“系统工具”下的 “终端”命令中通过指令来完成，切记不要使用任何微软操作系统下的工具来进行操作。

设计过程

Linux 下安装 XAMPP 的步骤如下：

（1）在 Linux 操作系统下，选择“主菜单”/“系统工具”/“终端”命令。
（2）在命令模式下，首先进入系统的根目录。
（3）通过 mkdir 命令在根目录下创建一个 opt 目录。
（4）通过 tar xvfz 命令将 xampp 解压缩到 opt 目录下。
（5）按 Enter 键，执行 XAMPP 的解压缩，直到安装成功，其具体使用的命令如图 1.72 所示。
（6）安装成功后，查看/opt/lampp 目录，如图 1.73 所示。

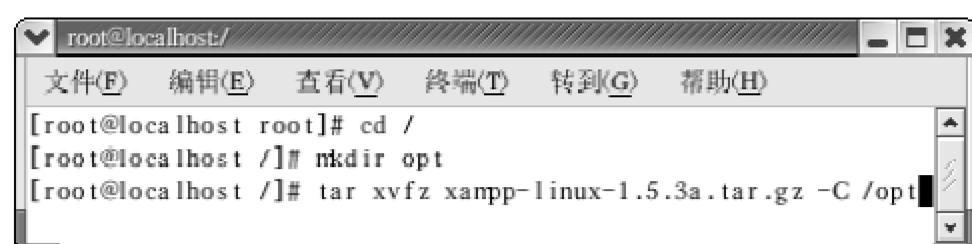


图 1.72　XAMPP 的安装



图 1.73　查看安装的内容

秘笈心法

心法领悟 016：Linux 下安装 XAMPP 的注意事项。

在 Linux 操作系统下执行 XAMPP 的解压缩时，必须进入到系统的根目录下，并且要在根目录下创建一个子目录，然后才可以执行解压缩的命令。否则，将提示找不到指定的目录或者文件。

实例 017　Linux 操作系统下启动 XAMPP

光盘位置：光盘\MR\01\017

中级

趣味指数：★★★★

实例说明

在实例 017 中，已经介绍了如何在 Linux 操作系统下安装 XAMPP，本实例我们将介绍在 Linux 操作系统下如何启动 XAMPP，其使用的命令如图 1.74 所示。

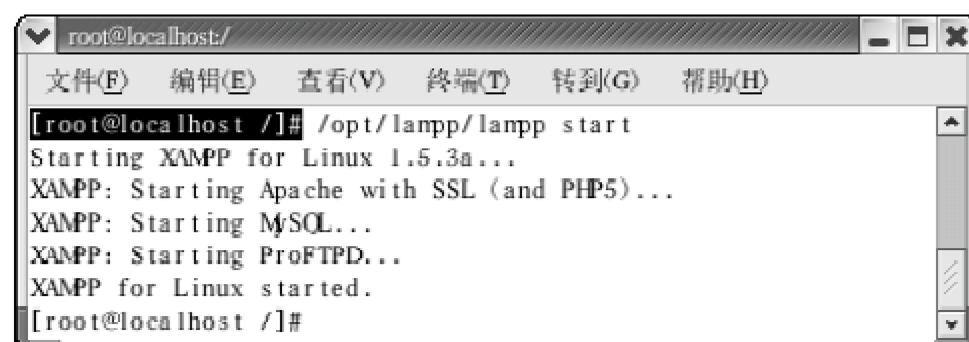


图 1.74　启动 XAMPP

关键技术

在 Linux 操作系统下启动 XAMPP，同样要进入到 Linux “主菜单”/“系统工具”下的 “终端”命令中，所使用的命令如下：

```
/opt/lamp/lamp start
```

设计过程

Linux 下启动 XAMPP 的步骤如下：

（1）在 Linux 操作系统下，选择“主菜单”/“系统工具”/“终端”命令。

（2）直接输入命令/opt/lamp/lamp start。

（3）按 Enter 键，执行 XAMPP 的启动操作。命令运行结果如图 1.75 所示。



图 1.75　XAMPP 启动成功

秘笈心法

心法领悟 017：停止 XAMPP 的命令。

Linux 下停止 XAMPP 的命令如下：

```
/opt/lamp/lamp stop
```

实例 018　设置 MySQL 数据库 root 用户的密码

中级

光盘位置：光盘\MR\01\018

趣味指数：★★★★☆

实例说明

在 XAMPP 中，默认情况下通过 phpMyAdmin 以 root 用户的身份登录 MySQL 数据库是不需要密码的，为了确保数据库的安全，应该给 root 用户设置一个密码。这就是本实例要介绍的内容——如何为 root 用户设置密码。

关键技术

在 Linux 操作系统下为 MySQL 数据库的 root 用户设置密码使用的命令如下：

```
/opt/lamp/bin/mysqladmin –u root password “newpassword”
```

设计过程

Linux 下设置 MySQL 数据库 root 用户密码的步骤如下：

（1）在 Linux 操作系统下，选择“主菜单”/“系统工具”/“终端”命令。

（2）直接输入命令/opt/lamp/bin/mysqladmin –u root password "111"，这里设置 root 用户的密码为“111”。

（3）按 Enter 键，执行该命令，其运行结果如图 1.76 所示。



图 1.76　设置 root 用户的密码

秘笈心法

心法领悟 018：卸载 XAMPP 命令。

Linux 下卸载 XAMPP 的命令如下：

```
rm –rf /opt/lampp
```

实例 019	第 5 个 PHP 程序 光盘位置：光盘\MR\01\019	中级 趣味指数：★★★★☆

实例说明

Linux 系统下 PHP 的集成开发环境 XAMPP 已经配置成功，下面编写一个小实例，应用 phpinfo()函数输出 PHP 的配置信息，通过这个实例熟悉一下在 Linux 下 XAMPP 的运用。在 Mozilla 浏览器的地址栏中输入 http://127.0.0.1/text/index.php，运行结果如图 1.77 所示。

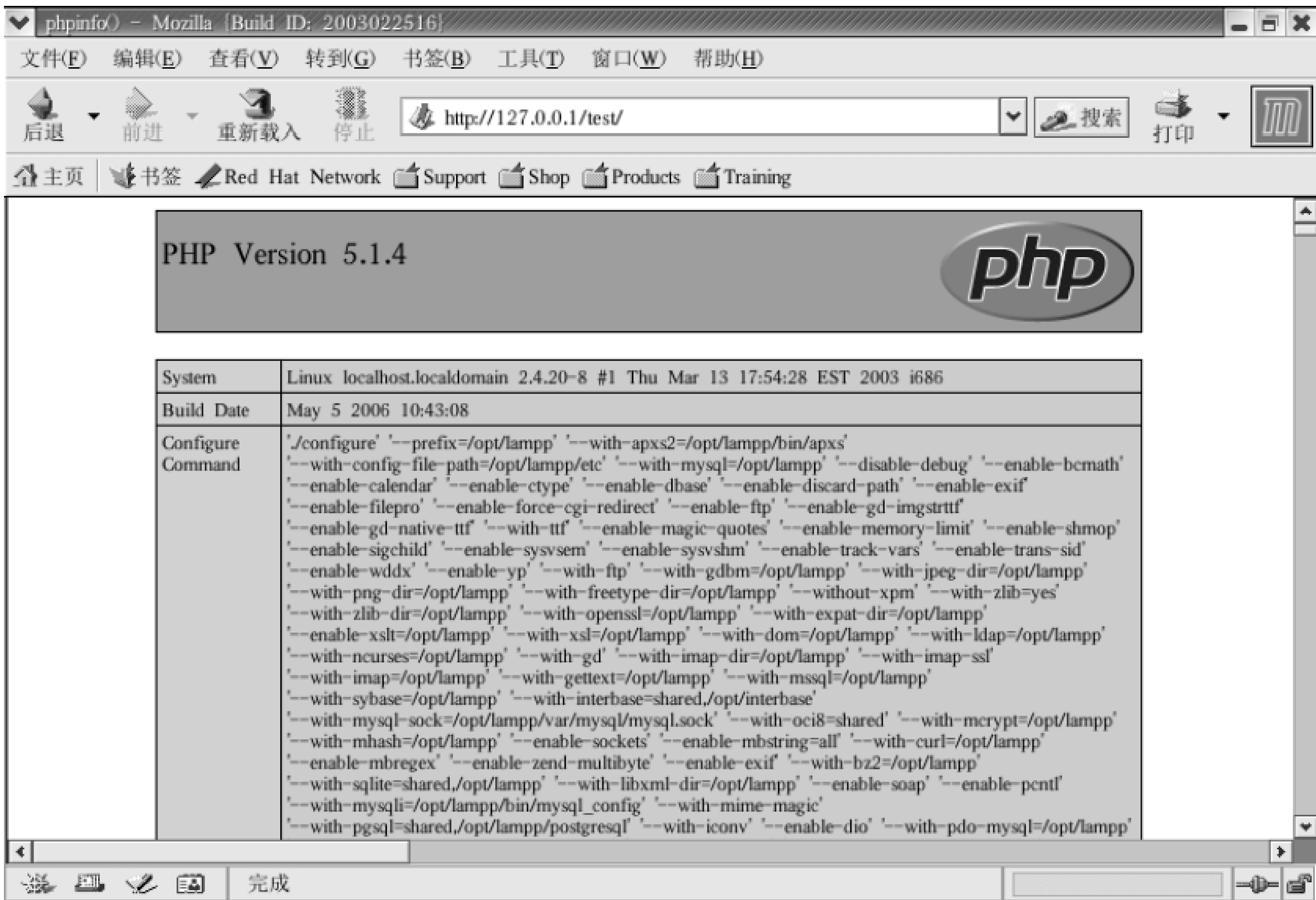


图 1.77 输出 PHP 的配置信息

关键技术

在 Linux 操作系统下运用 XAMPP 开发 PHP 程序，首先要确立 XAMPP 的程序运行文件夹，即明确创建的程序要存储在哪个文件夹中才可以运行。XAMPP 中的重要档案和目录如下：

☑ /opt/lampp/bin/：XAMPP 指令的根目录。例如 /opt/lampp/bin/mysql 用来执行 MySQL。
☑ /opt/lampp/htdocs/：Apache 文件根目录。
☑ /opt/lampp/etc/httpd.conf：Apache 设定档案。
☑ /opt/lampp/etc/my.cnf：MySQL 设定档案。
☑ /opt/lampp/etc/php.ini：PHP 设定档案。
☑ /opt/lampp/phpmyadmin/config.inc.php：phpMyAdmin 设定档案。

设计过程

在 XAMPP 中开发 PHP 程序的操作步骤如下：

（1）首先通过 gedit 工具编写一个 PHP 脚本，应用 phpinfo()函数输出 PHP 的配置信息，如图 1.78 所示。

（2）单击图 1.78 中的“文件”按钮，选择“另存为”命令，将该文件存储于 XAMPP 的 htdocs/目录下，

并且命名为 index.php，如图 1.79 所示。



图 1.78　编辑 PHP 脚本文件

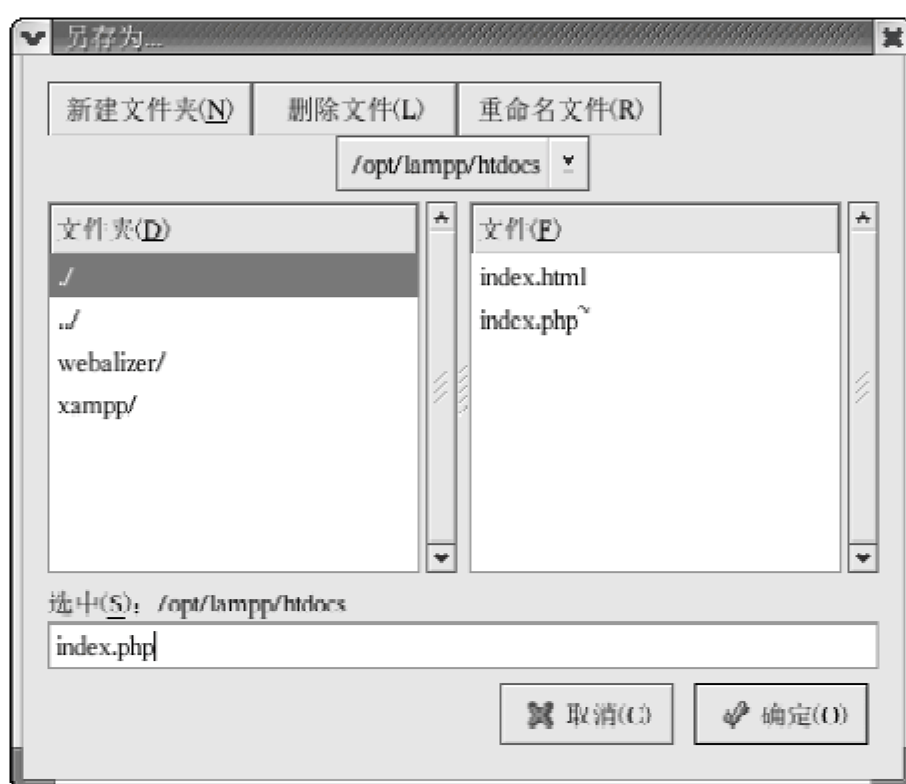


图 1.79　完成 PHP 文件的存储

（3）在 Mozilla 浏览器的地址栏中输入 http://127.0.0.1/text/index.php，运行结果如图 1.77 所示（参看“实例说明”）。

秘笈心法

心法领悟 019：Linux 下如何修改 MySQL 缺省字符集。

在 Linux 下修改 MySQL 缺省字符集为 utf8，操作的文件是 my.cnf，操作的内容如下：

```
[client]
default_character_set=utf8
[mysqld]
# default-character-set=utf8
character-set-server = utf8
collation-server = utf8_general_ci
[mysql]
default_character_set=utf8
```

1.6　Dreamweaver 开发工具

Dreamweaver 是 Macromedia 公司开发的 Web 站点和应用程序的专业开发工具。它将可视布局工具、应用程序开发功能和代码编辑组合在一起。

实例 020	Dreamweaver 中编码格式的选择 光盘位置：光盘\MR\01\020	中级 趣味指数：★★★★☆

实例说明

应用 Dreamweaver 开发网站，首先必须考虑网页的编码格式的选择。因为如果网页的编码格式有局限性，那么将导致网站在一些特定的情况下运行时出现乱码，将不利于网站程序的后期更新和维护。例如，如果网站在编写时使用的是 gb2312 编码，当程序在繁体操作系统中运行时就会出现乱码，结果如图 1.80 所示。

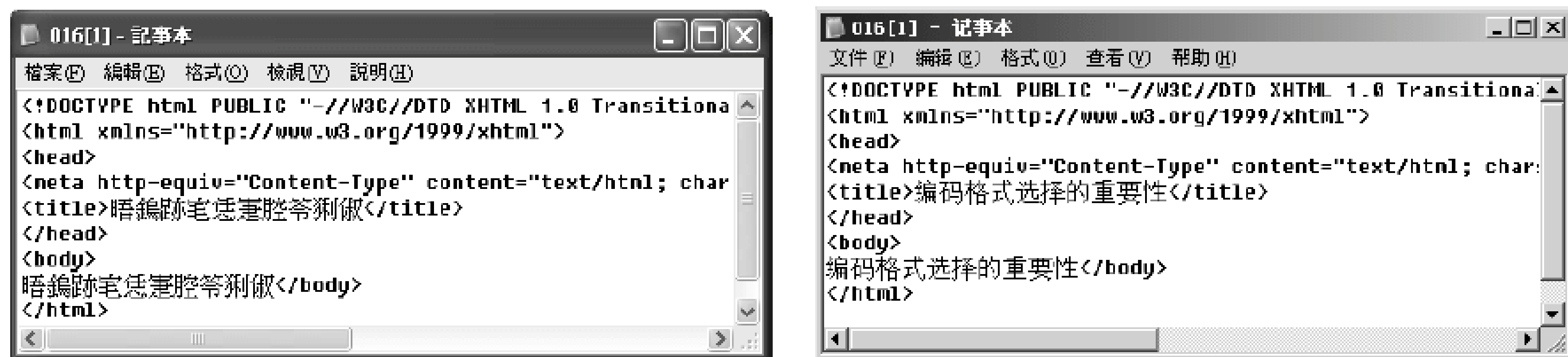


图 1.80　繁体操作系统下查看 gb2312 编码格式文件与源文件对比

关键技术

对于网站的开发，编码格式的选择很重要。如果使用 Dreamweaver 开发网站，那么可以在 Dreamweaver 编辑菜单的"首选参数"/"新建文档"/"默认编码"中设置新建文件的编码格式。这样就不必再为创建的每个文件的编码而担心，因为它们的编码格式是统一的。

设计过程

统一 Dreamweaver 创建文件编码格式的方法如下：

（1）打开 Dreamweaver 开发工具，选择"编辑"菜单命令，单击"首选参数"选项，将弹出如图 1.81 所示的页面。

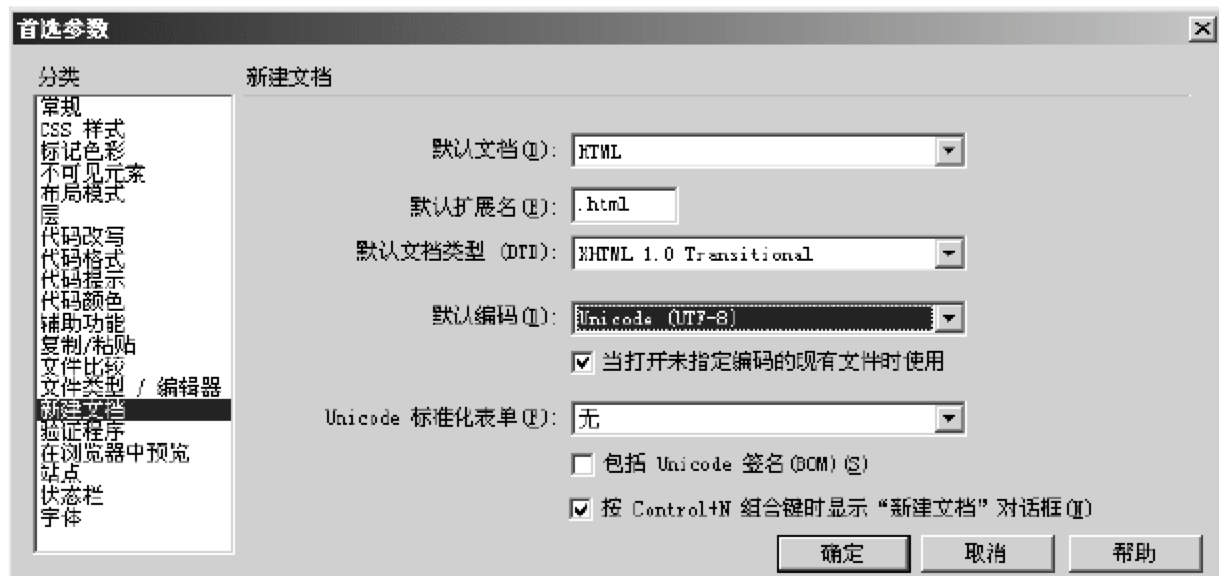


图 1.81　设置文件的编码格式

（2）在图 1.81 所示页面中，指定默认编码，终后单击"确定"按钮。

秘笈心法

心法领悟 020：在 Dreamweaver 中设置编码。

在 Dreamweaver 的"首选参数"设置中，不但可以设置默认编码的格式，还可以进行其他的一些设置，例如 css 样式、站点和字体等。

对于文件编码格式的选择，强烈建议读者使用 utf-8，因为这样可以与国际接轨，如果单纯地使用 gb2312 编码，那么一旦程序要更改编码格式，网页将出现乱码。如果使用 utf-8 编码格式，就不会出现任何问题，因为这个编码格式是通用的。

实例 021	Dreamweaver 中创建表格 光盘位置：光盘\MR\01\021	中级 趣味指数：★★★★

实例说明

讲解了在 Dreamweaver 中如何设置编码格式之后，下面介绍一下它的具体应用——在 Dreamweaver 中创建

表格。运行结果如图 1.82 所示。



图 1.82　在 Dreamweaver 中创建表格

关键技术

在 Dreamweaver 中创建表格，有两种方法：

第一种，选择菜单中的“插入”命令，在弹出的列表中选择“表格”命令，在弹出的“表格”对话框中完成表格的创建操作；

第二种，直接在“常用”工具栏中单击“表格”按钮，在弹出的“表格”对话框中完成表格的创建操作。

设计过程

在 Dreamweaver 中创建表格的操作步骤如下：

（1）打开 Dreamweaver 开发工具，选择“文件”/“新建”命令，在弹出的“新建文档”对话框中创建一个“动态页”/“PHP”文件，最后单击“创建”按钮，完成动态 PHP 文件的创建，如图 1.83 所示。



图 1.83　创建 PHP 动态文件

（2）如图 1.84 所示，在创建的 Untitled-1 文件中，首先切换到设计模式下，然后在工具栏中选择“常用”选项，单击“表格”按钮，在弹出的“表格”对话框中完成表格的创建。



图 1.84　在设计模式下添加表格

（3）表格创建对话框如图 1.85 所示，在其中可以设置表格的行数、列数、表格宽度、边框粗细、单元格边距和单元格间距等。设置完成后单击“确定”按钮，完成表格的创建。



图 1.85　表格的创建

（4）表格创建完成后，将在设计界面中输出如图 1.86 所示的内容。

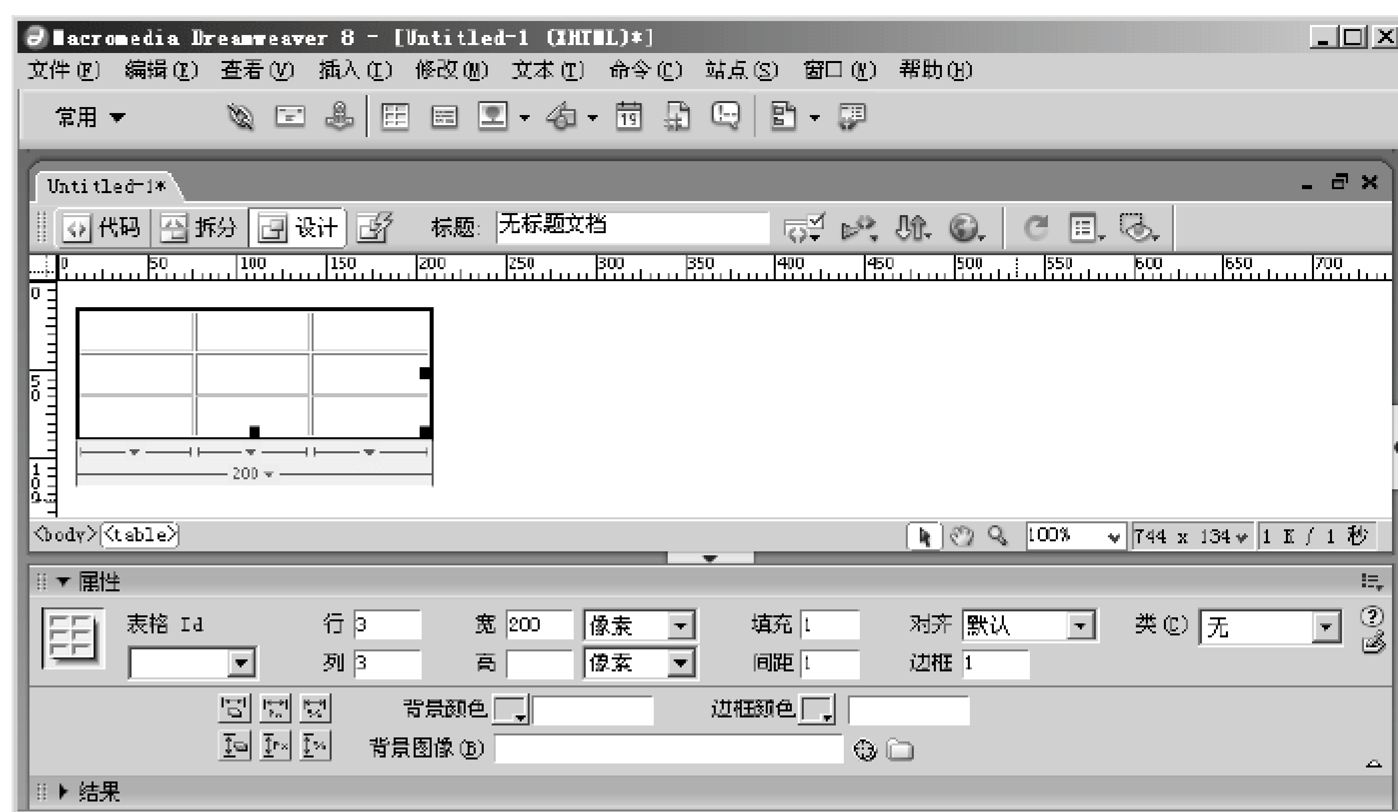


图 1.86　创建完成的表格

（5）表格创建完成，选择“文件”/“保存”命令，将其存储于 apache 服务器指定的文件夹下，命名为 index.php。运行结果请参考实例说明。

秘笈心法

心法领悟 021：操作表格的小技巧。

在图 1.86 所示的表格设计界面中，表格处于选中状态时，可以设置其属性，包括行数、列数、表格宽度、边框粗细、单元格边距、单元格间距、背景颜色、边框颜色和背景图像等。

实例 022　在表格中插入宠物照片

光盘位置：光盘\MR\01\022

中级

趣味指数：★★★★☆

实例说明

实例 021 讲解如何在 Dreamweaver 中创建表格，本实例将讲解如何向表格中插入图片。本实例的运行结果如图 1.87 所示。



图 1.87　表格中插入宠物图片

关键技术

在 Dreamweaver 中，向创建的表格中插入图片时，必须注意图片的存储位置，最好的方式就是先将图片存储到实例的根目录下，这样当实例在运行过程中，不会因为机器的更换而找不到图片。

如果没有将图片存储到实例的根目录下，在执行图片的插入操作过程中，将给出一个如图 1.88 所示的提示信息，提示是否将图片存储到实例的根目录下。

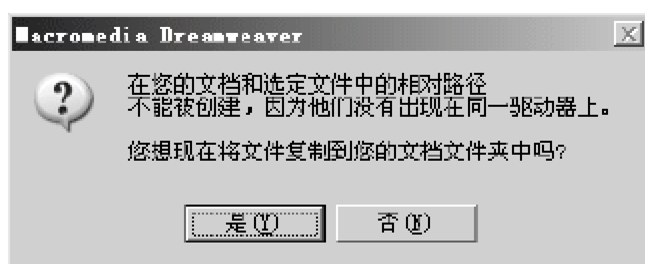


图 1.88　插入不在同一驱动器上的文件时给出的提示信息

当单击“是”按钮时，即可将图片复制到实例的根目录下。但是，这种做法有一个弊端，就是文件的存储没有规范性。试想一下，如果网页中需要插入多个图片，程序的根目录下将会是一个什么景象？所以最好的方法是先将要插入的图片存储到实例根目录下的一个指定文件夹中，这样做比较规范，也容易管理，一般都将这个图片存储文件夹命名为 images。

如果在图 1.88 的提示信息中，单击“否”按钮，虽然图片仍然可以插入到表格中，但在查看网页代码时会发现，img 标签中 src 属性指定的图片文件路径是本机的绝对路径（file:///E|/Program Files/ 9.gif），在本机中仍然可以运行，但要将本实例复制到其他机器上运行的话，在指定的路径下将找不到图片文件。

设计过程

在 Dreamweaver 中创建表格的操作步骤请参考实例 021，这里不再赘述，下面介绍如何在创建好的表格中插入图片。

（1）首先，将要使用的图片复制到本实例根目录下的 images 文件夹中。

（2）将光标定位到指定要插入图片的单元格中，单击 Dreamweaver 开发工具中的“插入”按钮，在弹出的列表框中选择“图像”命令，在弹出的“选择图像源文件”对话框中，选择要插入的图片，如图 1.89 所示。



图 1.89　插入图片

（3）确定要插入的图片之后，在图 1.89 中单击“确定”按钮，即完成图片的插入操作。

（4）保存文件，在 IE 浏览器中输入网址，即可运行本实例，运行结果请参看实例说明。

秘笈心法

心法领悟 022：操作 Dreamweaver 小技巧。

在 Dreamweaver 的设计模式下，当选中插入的图片时，在开发工具的下方将出现图像的属性对话框，如图 1.90 所示。通过这个属性对话框可以设置图像的宽度、高度、链接以及是否添加地图等。



图 1.90　图像的属性操作界面

实例 023　Dreamweaver 创建表单

光盘位置：光盘\MR\01\023

中级

趣味指数：★★★★☆

实例说明

本实例讲解如何在 Dreamweaver 中创建表单。表单是网站与客户沟通的一座桥梁，通过它可以直接将客户的信息反馈给网站的管理者，达到企业与网站浏览者更好的交互效果。在本实例中将对在 Dreamweaver 中创建表单进行详细讲解，本实例的运行结果如图 1.91 所示。



图 1.91　创建表单元素

关键技术

在 Dreamweaver 中创建表单，首先要在工具栏中选择“表单”选项，然后就可以通过不同的按钮创建不同的表单元素，其中每个按钮对应的功能如表 1.1 所示。

表 1.1　Dreamweaver 中的表单元素

图　像	名　称	说　明
	表单	<form name="form2" method="post" action=""></form> name：表单的名称 method：表单提交的方法，包括 POST 和 GET 方法 action：表单提交的路径

续表

图　像	名　称	说　明
	文本字段	<input type="text" name="textfield"> type：应用表单的类型 name：文本框的名称
	隐藏域	<input type="hidden" name="ID" value=""> type：表单的类型，其中的 hidden 表示隐藏域 name：隐藏域的名称，可以自己定义 value：隐藏域的值，可以填写隐藏域的默认值
	文本区域	<textarea name="" cols="" rows="" id=""></textarea> <textarea>…</textarea>，文本域的标记 name：文本域的名称，例如其中的“test” cols：表示文本域字符的宽度 rows：表示有多少行字符 初始值在<textarea></ textarea>标记之间进行输入，例如其中的“欢迎大家访问我们的论坛”
	复选框	<input type="checkbox" name="checkbox" value="体育"> type：表单的类型，其中的“checkbox”表示复选框 name：复选框的名称，例如其中的 checkbox value：复选框提交的值，例如其中的体育 checked：如果希望预先为用户勾选某些选项，可以为这些选项加上 checked 参数 disable：如果希望某一个选项失效，可以加上 disabled 参数
	单选按钮	<input type="radio" name="radiobutton" value="radiobutton" checked="checked" />男 <input type="radio" name="radiobutton" value="radiobutton" />女
	单选按钮组	<input type="radio" name="RadioGroup1" value="单选" />单选 <input type="radio" name="RadioGroup1" value="单选" />单选
	列表\菜单	<select name="select2"> <option selected="selected">默认值</option> <option value="对应值 1">列表值 1</option> <option value="对应值 2">列表值 2</option> ...... </select> name：指该<select>组件的名称 option：提供给用户选择的项目，其中的 value 是该选项所代表对应的选择值，可以省略
	跳转菜单	<select name="menu1" onchange="MM_jumpMenu('parent',this,0)"> <option>unnamed1</option> </select> 跳转菜单，通过表单实现指定网址之间的跳转
	图像域	<input type="image" name="imageField" src="images/QQ.gif" />在表单中插入图片
	文件域	<input type="file" name="file" />完成文件的提交
	按钮	<input type="submit" name="Submit" value="提交" />创建的提交按钮，如果 stype 的值为 button，那么它表示一个普通的按钮，不具备提交的功能 <input type="reset" name="Submit2" value="重置" />创建的重置按钮
	标签	<label>标签</label>
	字段集	<fieldset><legend>字段集</legend></fieldset>

设计过程

在 Dreamweaver 中创建表单的操作步骤如下：

（1）新建一个动态 php 文件，切换到设计模式下。

（2）在工具栏中选择“表单”命令，如图 1.92 所示。

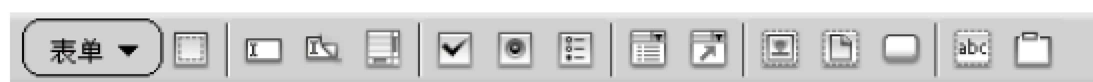


图 1.92　选择表单工具

（3）添加一个 form 表单，如图 1.93 所示。

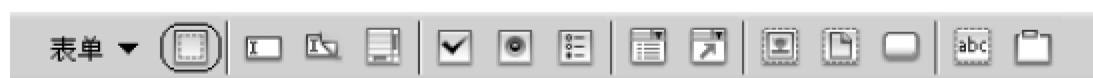


图 1.93　添加表单

（4）根据实际需要，添加不同的表单元素，并设置相应的名称和值，如图 1.94 所示。

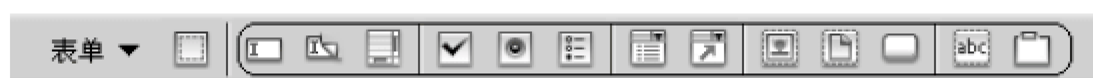


图 1.94　添加表单元素

最终创建的表单元素设计效果如图 1.95 所示。

会员名称	
简介：	请输入您的个人信息
个人爱好：	☑ 音乐 ☐ 体育 ☐ 文学
性别：	○ 男 ○ 女
语言选择：	○ PHP ○ Java ○ VB
图书选择：	Java开发实战宝典
头像选择：	浏览...
提交 重置	

图 1.95　表单元素的设计效果

秘笈心法

心法领悟 023：通过表单传值时的注意事项。

在通过表单中的元素传递值的时候，一定要正确地书写表单元素的名称，其中不应该有空格存在；在获取表单元素的值时，表单元素的名称一定要与 form 中设置的名称相同，同时还要注意大小写的统一，否则将不能获取到表单元素的值。

实例 024　Dreamweaver 中创建和附加 CSS 样式

光盘位置：光盘\MR\01\024

中级

趣味指数：★★★★☆

实例说明

本实例讲解如何在 Dreamweaver 中创建和附加 CSS 样式。通过 CSS 能够更好地对网页中的表格、文本和图像的样式进行控制，并且使其更易于维护和更新。本实例的运行结果如图 1.96 所示。

编程词典宣传语总结
（1）计算机语言中的“新华字典”。具备海量的数据资源，帮助您解决在学习编程语言中遇到的问题。（知识）
（2）编程者的良师益友。为您的学习提供完整的学习方案，给您提供理论和实践相结合的机会。（学习方法、实例、实践练习）
（3）拿来主义的倡导者。编程词典中提供了完整、丰富的页面、程序和软件资源，可以作为参考，也可以直接使用。（方案、程序）
（4）完美的售后服务和交互方式。QQ、电话、EMAIL

图 1.96　Dreamweaver 中创建和附加 CSS 样式

关键技术

在 Dreamweaver 中创建 CSS 样式，必须明确新建 CSS 样式的规则，如图 1.97 所示。

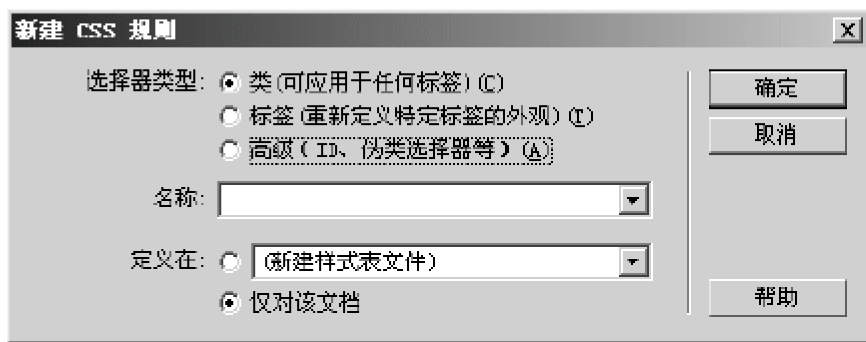


图 1.97　新建 CSS 规则

如果选择的是“新建样式表文件”选项，那么定义的 CSS 存储于指定的文件夹下；如果选择的是“仅对该文档”选项，那么定义的 CSS 存储于该文件中。

如果只是对单独的某个文件进行操作，建议将 CSS 定义到当前文件中，但是，如果是在网站中应用 CSS，那么应该将 CSS 样式定义到单独的文件中，并且以类的形式进行定义，因为这样便于对 CSS 样式的修改，使网站的样式更加统一，而且可以避免代码的冗余。

设计过程

创建一个名称为 styles 的样式，将其定义到新建样式表中，使用类的形式进行定义，其具体的操作步骤如下：

（1）在 Dreamweaver 的设计模式下，单击右键选择“CSS 样式”/“新建”命令，在弹出的“新建 CSS 规则”对话框中定义 CSS 样式的规则，如图 1.98 所示。

（2）如图 1.98 所示，CSS 样式的规则定义完成，单击“确定”按钮，弹出“保存样式表文件为”对话框，定义 CSS 样式文件的存储位置，如图 1.99 所示。

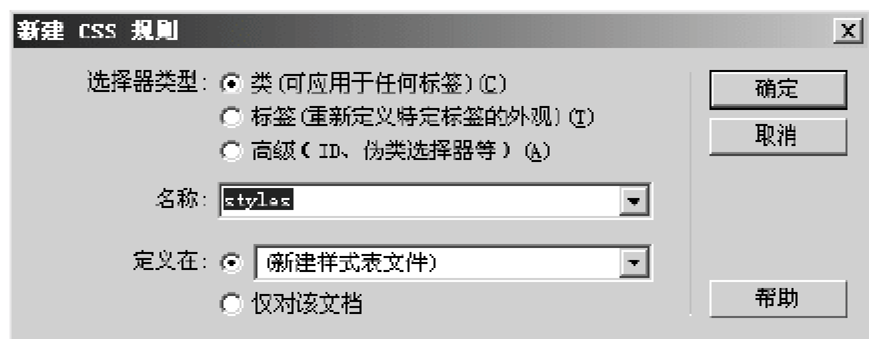


图 1.98　定义 CSS 样式的规则



图 1.99　定义 CSS 样式表的存储位置

（3）确定存储位置和名称后，单击“保存”按钮，将弹出 styles 样式具体的规则定义对话框，如图 1.100 所示。在其中，可以根据“分类”，对类型、背景、区块、方框、边框、列表、定位和扩展分别进行定义，单击“确定”按钮，styles 样式创建完成。



图 1.100　定义 styles 样式的规则

（4）在 Dreamweaver 的设计模式下，选中被操作的内容，单击右键选择“CSS 样式”命令，即可选择指定的样式文件，如图 1.101 所示。

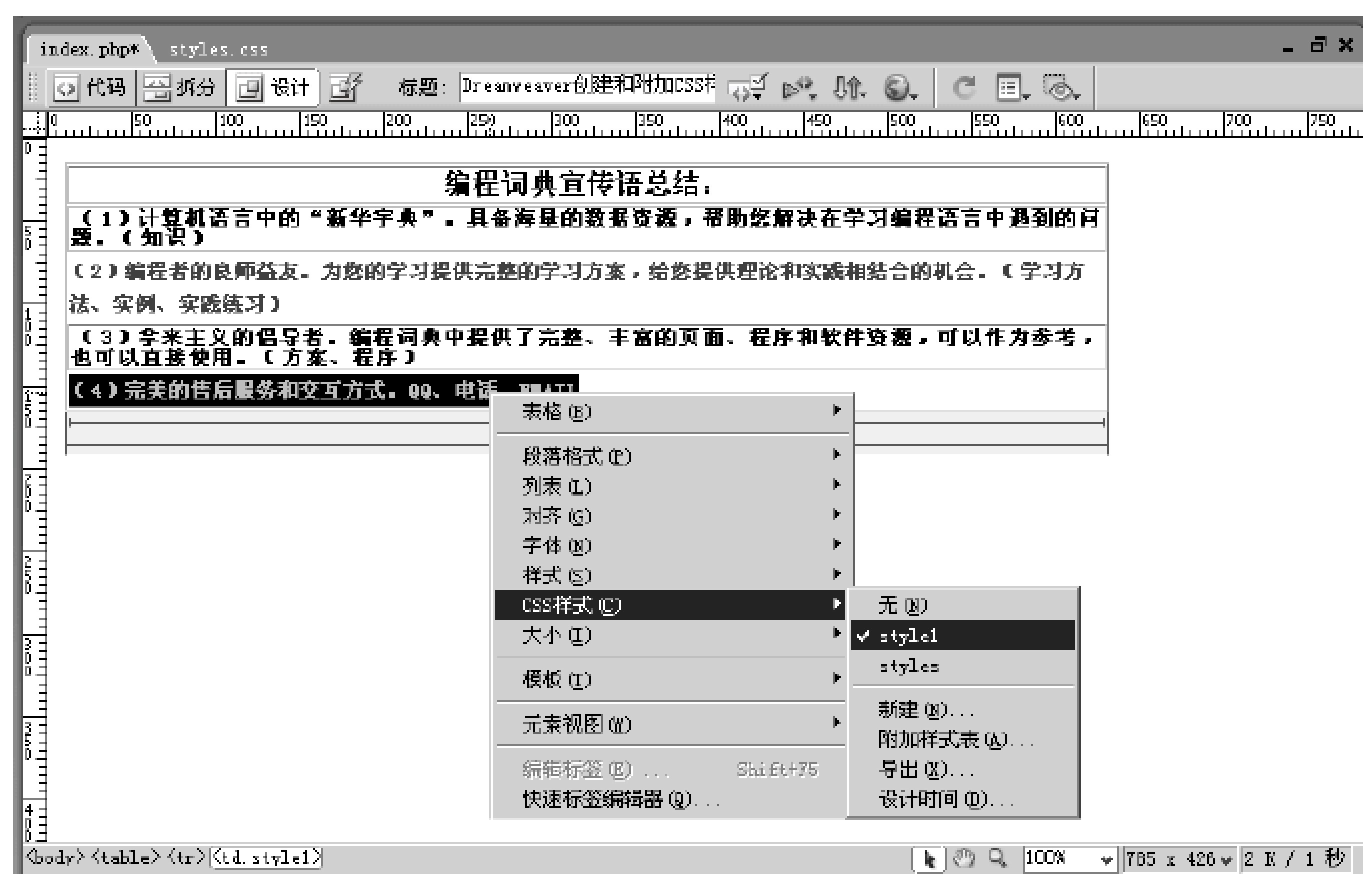


图 1.101　应用定义的 CSS 样式

✍ **技巧：**附加 CSS 样式文件。在 Dreamweaver 的设计模式下，单击右键选择“CSS 样式”/“附加样式表”命令，在弹出的“链接外部样式表”对话框中，选择定义在指定文件夹下的 CSS 样式文件。如果选择“链接”则直接通过 link 标签链接到 CSS 样式文件；如果选择导入，则通过 @import url 导入指定的 CSS 文件。确立附加的 CSS 样式文件后，单击“确定”按钮。

■ 秘笈心法

心法领悟 024：在 Dreamweaver 中创建 CSS 样式的相关说明。

在 Dreamweaver 中创建的 CSS 样式，可以在代码模式下对其进行修改，也可以在设计模式下对其进行修改，如图 1.102 所示。

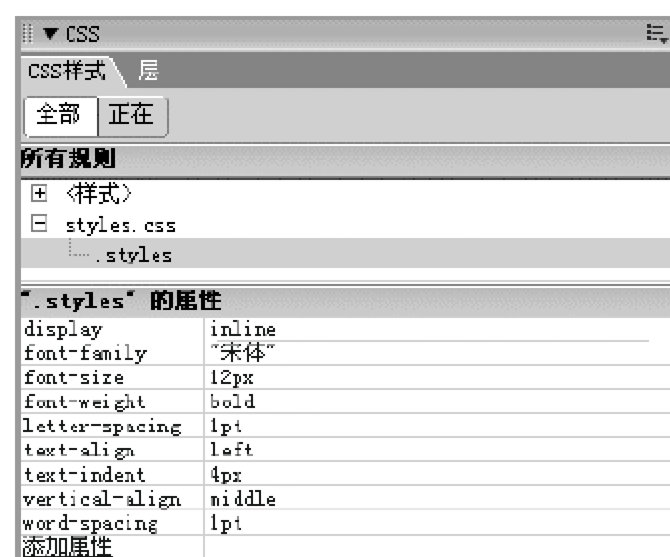


图 1.102　在设计模式下操作 CSS 样式

实例 025　Dreamweaver 控制弹出信息

光盘位置：光盘\MR\01\025

中级

趣味指数：★★★★☆

实例说明

在 Dreamweaver 中不但可以创建表格、表单或者 CSS 样式等内容，而且可以通过行为，实现一些特殊的功能。例如，在本实例中，定义一个“弹出信息”行为，当用户单击指定的图片时，将弹出一个提示对话框，如图 1.103 所示。



图 1.103　通过 Dreamweaver 直接创建的弹出信息

关键技术

在 Dreamweaver 中控制弹出信息的原理是自动创建 JavaScript 脚本文件，定义方法，然后通过 onclick 调用脚本中的方法，完成当鼠标单击指定的内容时弹出信息。

JavaScript 脚本中定义的方法如下：

```
<script type="text/JavaScript">
<!--
function MM_popupMsg(msg) { //v1.0
  alert(msg);
}
//-->
</script>
```

在表单的图像域中，通过 onclick 事件完成对脚本文件方法的调用，其代码如下：

```
<input name="imageField" type="image" onclick="MM_popupMsg('请输入完整的注册信息')" src="images/09.jpg" />
```

设计过程

在 Dreamweaver 中添加“弹出信息”行为的操作步骤如下：

（1）通过 Dreamweaver 新建一个 index.php 文件，创建一个用户注册页面。

（2）在 Dreamweaver 的设计模式下，选择用户注册的图像域，然后选择“窗口”/“行为”命令，在 Dreamweaver 操作界面的右侧将弹出如图 1.104 所示的内容。

（3）在图 1.104 所示的页面中，单击“+”的下标，在弹出的列表框中选择“弹出信息”命令，将弹出如图 1.105 所示的“弹出信息”编辑页面，在“消息”文本框中定义弹出信息的内容，最后单击“确定”按钮，

弹出信息的行为创建完毕。



图 1.104　行为控制面板



图 1.105　编辑弹出信息

秘笈心法

心法领悟 025：在 Dreamweaver 中实现一些 JavaScript 功能。

在 Dreamweaver 中，“弹出信息”只是众多行为中的一个，还可以通过“窗口”中的行为实现很多的功能，如图 1.106 所示。



图 1.106　行为的功能展示

实例 026　Dreamweaver 控制浏览器的窗口　　中级

光盘位置：光盘\MR\01\026　　趣味指数：★★★★

实例说明

本实例将介绍 Dreamweaver 中行为的另一种功能——控制打开浏览器的窗口。通过该功能可以对打开的浏览器窗口进行设置，包括窗口的高度、宽度以及是否包含导航工具栏、地址工具栏和菜单栏等，本实例的运行结果如图 1.107 所示。



图 1.107　Dreamweaver 控制浏览器的窗口

关键技术

控制打开浏览器的窗口，同样是 Dreamweaver 中行为的一种功能，通过该功能可以控制网页窗口的打开模式，例如，网页窗口中导航条的使用、滚动条的使用、地址工具栏的应用以及网页窗口的大小等。

行为中“打开浏览器窗口”的操作界面如图 1.108 所示。

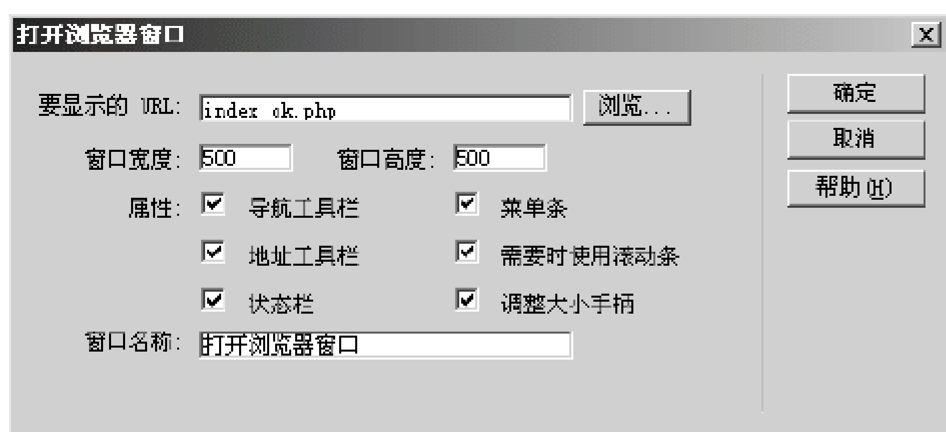


图 1.108　打开浏览器窗口

设计过程

在 Dreamweaver 中添加“打开浏览器窗口”行为的操作步骤与添加“弹出信息”行为是相同的，唯一的区别是在单击“+”的下标时，在弹出的列表框中选择“打开浏览器窗口”命令，将弹出如图 1.108 所示的“打开浏览器窗口”对话框。

秘笈心法

心法领悟 026：在 Dreamweaver 中屏蔽标准按钮。

通过“打开浏览器窗口”行为，我们可以屏蔽浏览器窗口中的标准按钮，这个功能在某些程序中是非常有用的，例如，在一些需要屏蔽刷新功能的网页中，通过“打开浏览器窗口”行为来实现是比较不错的方法。

实例 027　通过 Dreamweaver 创建站点

光盘位置：光盘\MR\01\027

中级

趣味指数：★★★★☆

实例说明

本实例讲解如何在 Dreamweaver 中创建站点，从而不必在 IE 浏览器中输入地址，只需按 F12 键，即可完成对所创建程序的浏览操作。

关键技术

在 Dreamweaver 中创建站点和配置测试服务器时，一定要注意将本地的 HTTP 地址与测试服务器中的 URL 前缀统一，即都指定到站点的根目录下。例如，指定 HTTP 的地址是 http://localhost//mr/01/，那么测试服务器的 URL 前缀也必须是 http://localhost//mr/01/或者 http://127.0.0.1//mr/01/。

设计过程

在 Dreamweaver 中创建站点的操作步骤如下：

（1）打开 Dreamweaver 开发工具，选择菜单栏中的“站点”/“新建站点”命令，在如图 1.109 所示的对话框中添加站点名称、设置本地根文件夹、链接相对于“站点根目录”、设置 HTTP 地址。

（2）在图 1.109 中，选择左侧的“测试服务器”选项，弹出如图 1.110 所示的“测试服务器”对话框，选择服务器模型“PHP MYSQL”，访问为“本地/网络”，测试服务器文件夹为“F:\PkhPHP\www\”，URL 前缀

为“http://localhost/”，最后单击“确定”按钮。

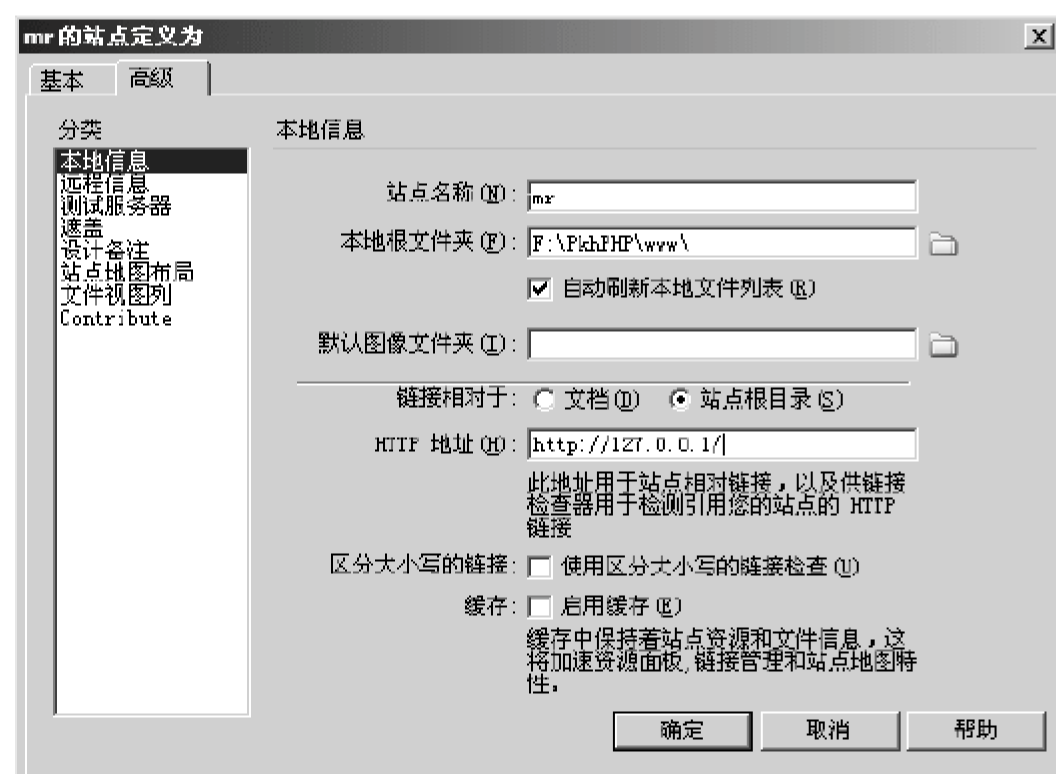


图 1.109　定义 mr 站点

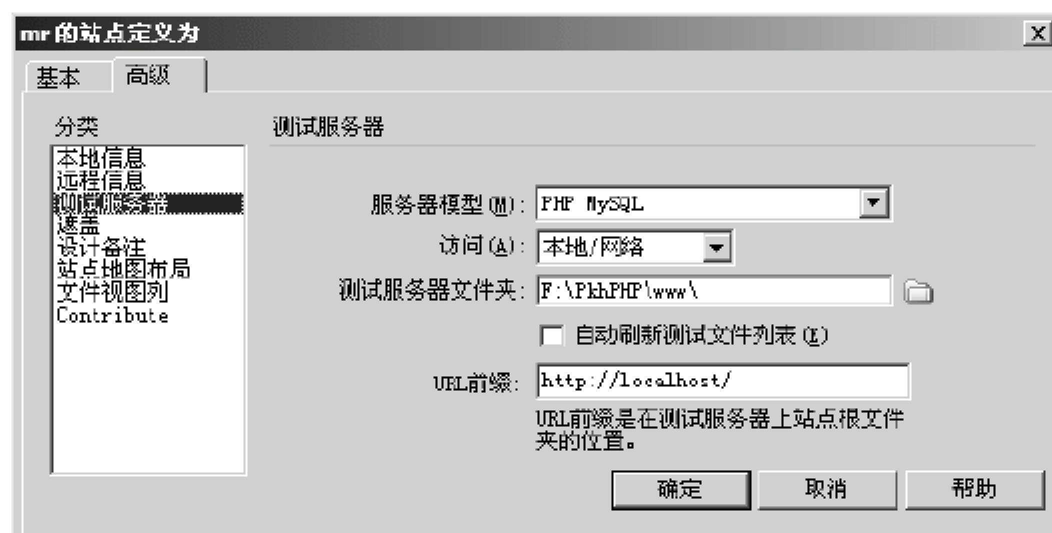


图 1.110　配置测试服务器

（3）mr 站点和测试服务器设置完毕后，就可以在 Dreamweaver 下直接使用快捷键 F12 来浏览程序了。

秘笈心法

心法领悟 027：在 Dreamweaver 中创建站点的注意事项。

在 Dreamweaver 中创建站点和配置测试服务器时，如果本地的 HTTP 地址与测试服务器中的 URL 前缀不统一，那么就不能够通过 F12 键直接浏览程序。

实例 028　通过 Dreamweaver 开发第 1 个 PHP 程序

光盘位置：光盘\MR\01\028

中级

趣味指数：★★★★☆

实例说明

前面实例讲解 Dreamweaver 自身功能的运用，在本实例中应用 Dreamweaver 开发一个最简单的 PHP 程序，即输出一段欢迎信息，其目的一是了解 PHP 的语法规则，二是熟悉 Dreamweaver 开发工具的使用。运行结果如图 1.111 所示。



图 1.111　第 1 个 PHP 程序

关键技术

必须将通过 Dreamweaver 开发工具编写的 PHP 脚本存储到 PHP 开发环境指定的服务器文件夹下才可以运行。

设计过程

（1）打开 Dreamweaver 开发工具，新建一个 PHP 项目，如图 1.112 所示。



图 1.112　新建一个 PHP 项目

（2）单击图 1.112 中的 PHP 项目图标，打开一个 PHP 项目文件夹，如图 1.113 所示。

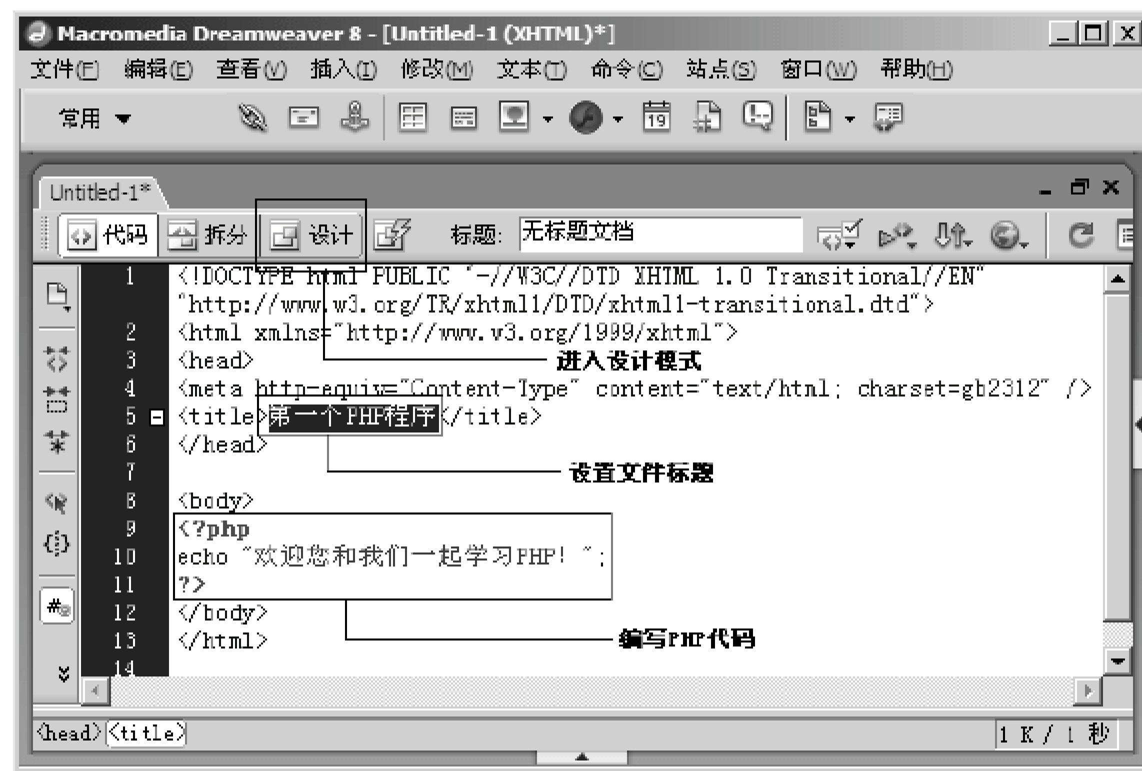


图 1.113　新的 PHP 项目文件

（3）在图 1.113 所示的文件中，首先定义文件的标题，<title>第一个 PHP 程序</title>，然后编写 PHP 代码，其代码如下：

```
<?php
echo "欢迎进入 PHP 的世界！！";
?>
```

PHP 代码分析：

☑　“<?php”和“?>”是 PHP 的标记对。在这对标记中的所有代码都被当作 PHP 代码来处理。

☑　echo 是 PHP 中的输出语句，与 ASP 中的 response.write、JSP 中的 out.print 含义相同，输出字符串或者变量值。每行代码都以分号“;”结尾。

（4）保存文件，单击图 1.113 中的“文件”按钮，选择“另存为”命令，将编写的文件保存在站点的根目录 www 文件夹中，并命名为 test.php，最后单击“保存”按钮，如图 1.114 所示。

■ 秘笈心法

心法领悟 028：运行程序的两种方法。

程序的运行有两种方法：第一种，直接在 Dreamweaver 开发工具中，按 F12 键运行程序；第二种，在 IE 浏览器的地址栏中输入 http://127.0.0.1/test.php。程序运行的前提是必须将程序存储到服务器指定的文件夹中。



图 1.114　保存编写的 PHP 文件

1.7　Zend Studio 开发工具

Zend Studio 是目前比较流行的 PHP 开发工具，囊括了编辑、调试、配置 PHP 程序所需要的客户及服务器组件，尤其是功能齐全的调试功能，让 PHP 错误不再棘手。

Zend Studio 也有不足之处，一是其运行速度慢；二是没有 Dreamweaver 的界面设计功能，只能通过浏览器来浏览页面设计效果；三是其是一款收费软件（但可以下载试用版），下载地址为 http://www.zend.com/store/products/zend-studio.php。

实例 029　安装 Zend Studio

光盘位置：光盘\MR\01\029

中级　趣味指数：★★★☆

实例说明

在应用 Zend Studio for Eclipse 之前，我们有必要介绍一下它的安装步骤，因为只有正确地完成安装，才能够顺利地应用。

关键技术

在安装 Zend Studio for Eclipse 的过程中，关键是安装内容的选择、安装位置的选择和定义 Zend 支持的文件格式，这是在安装过程中必须注意的内容。

设计过程

（1）运行安装文件，进行安装准备工作，如图 1.115 所示。

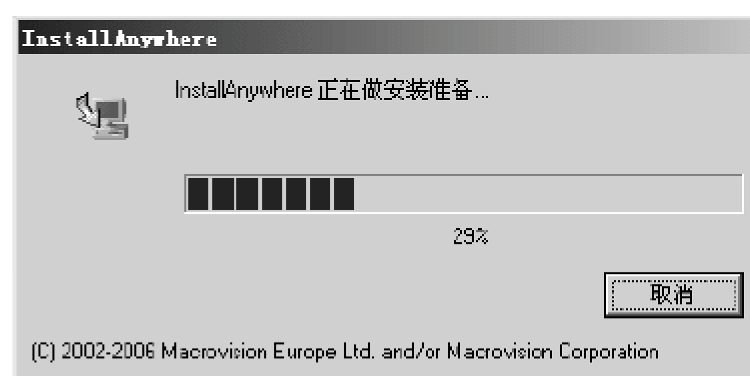


图 1.115　安装 Zend Studio for Eclipse 的准备工作

（2）进入到安装页面，如图 1.116 所示，单击 Next 按钮。

（3）阅读服务器条款，选择同意，如图 1.117 所示，单击 Next 按钮进入下一页面。

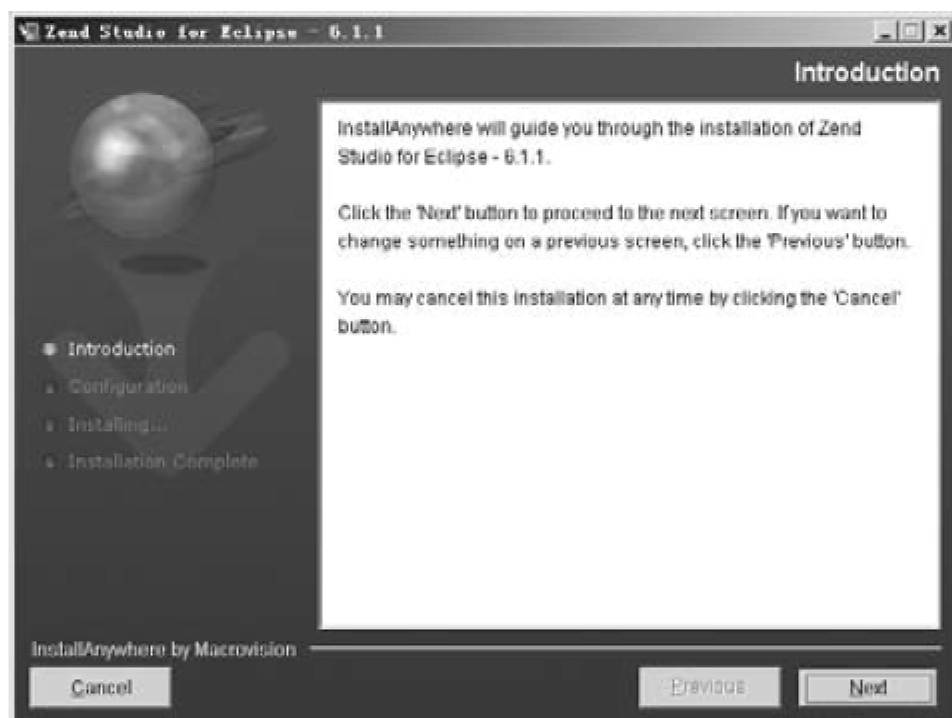


图 1.116　开始进行安装

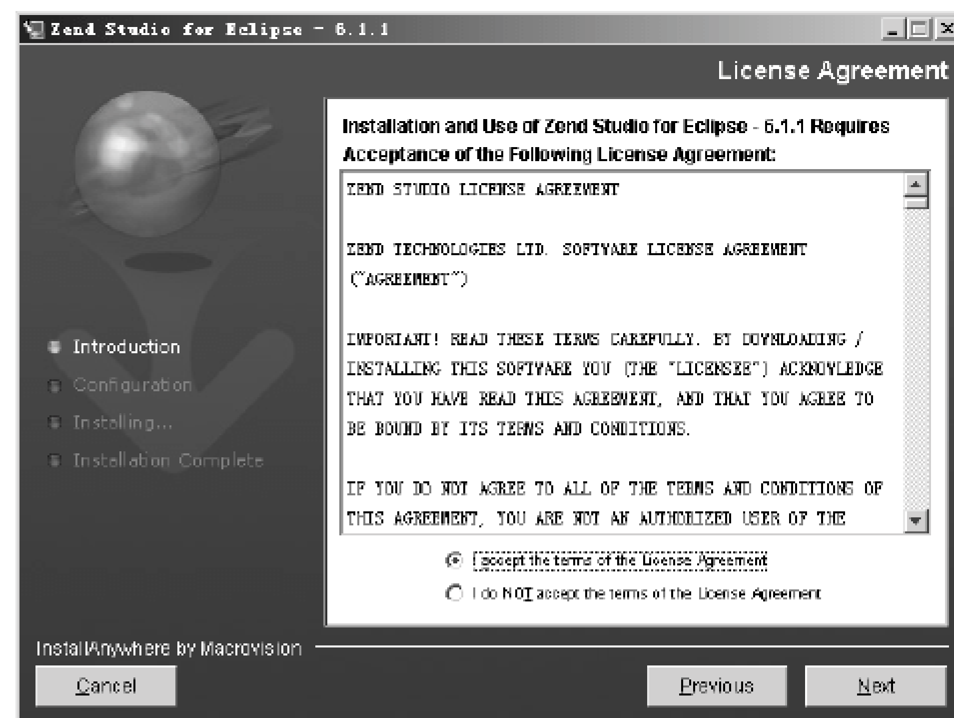


图 1.117　阅读服务条款

（4）选择安装内容，其中包括一个 Zend 浏览器和 Zend Studio，全部勾选，如图 1.118 所示。单击 Next 按钮，进入到下一页面。

（5）设置文件的安装目录，用户可以任意指定安装位置，确定单击 Next 按钮进行下一项操作，如图 1.119 所示。

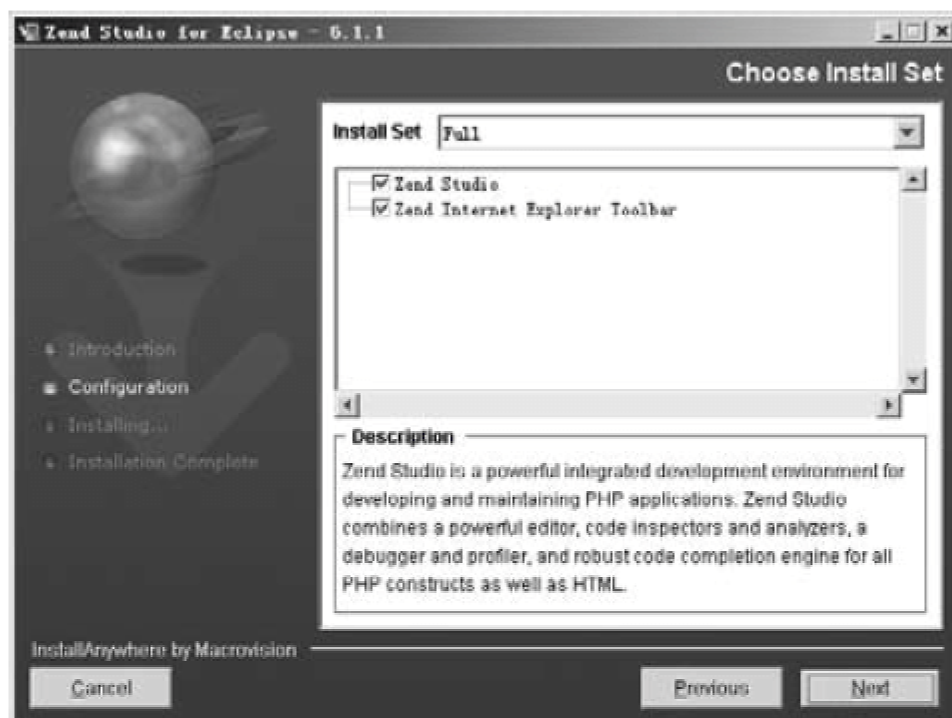


图 1.118　选择安装内容



图 1.119　设置工具安装位置

（6）定义 Zend 支持的文件格式，全部勾选，如图 1.120 所示。

（7）确认安装信息，如图 1.121 所示，单击 Install 按钮开始进行安装。

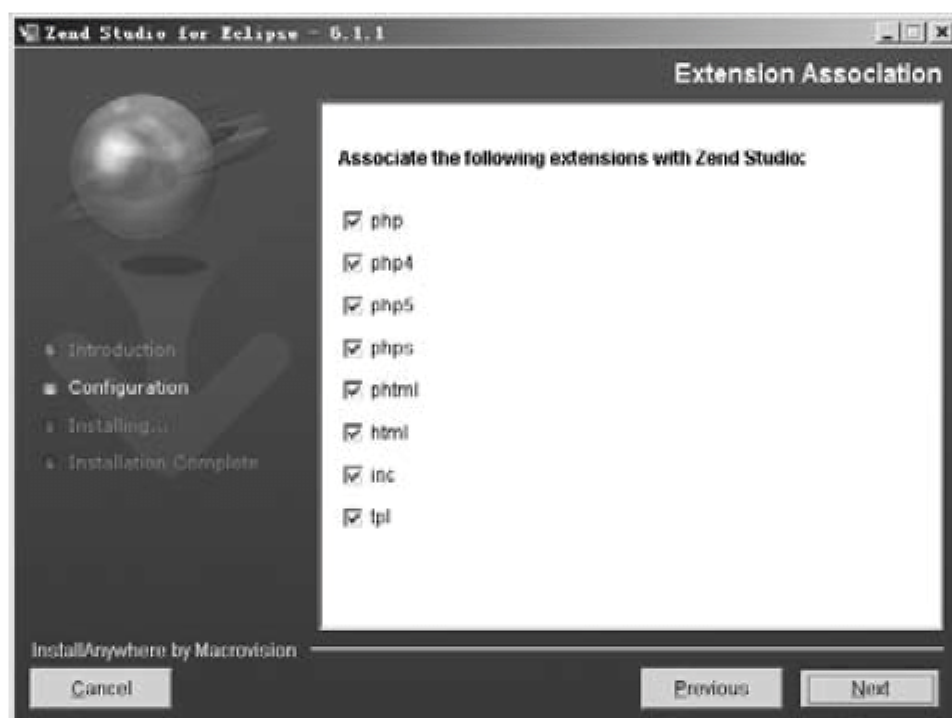


图 1.120　定义 Zend 支持的文件格式

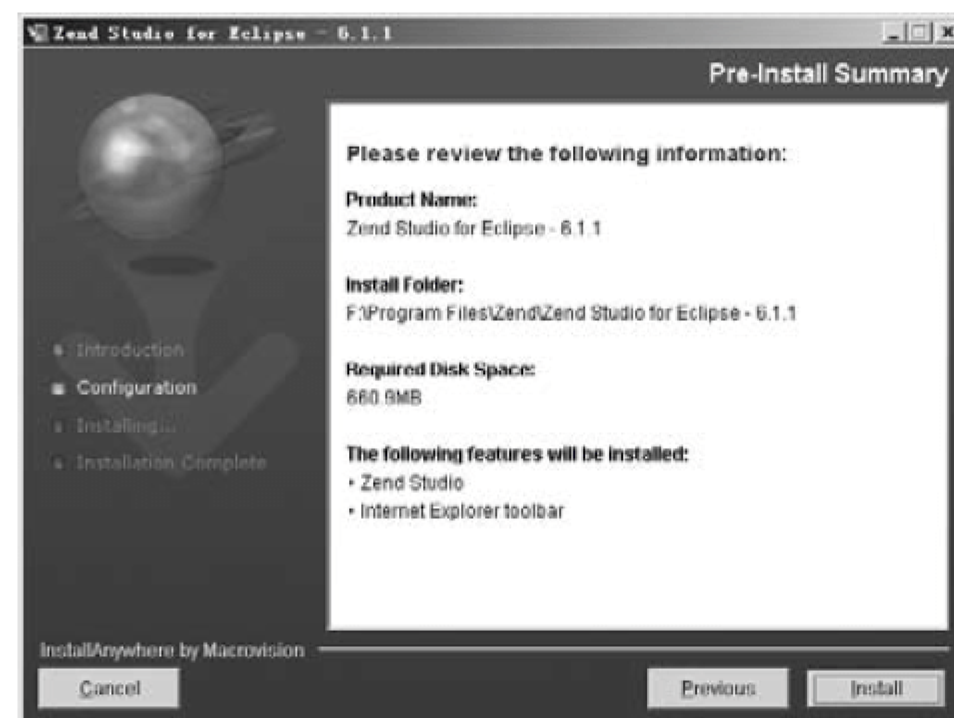


图 1.121　确认安装信息

（8）安装进行中，如图 1.122 所示。

（9）安装完成，单击 Done 按钮，如图 1.123 所示。



图 1.122 Zend 安装进行中

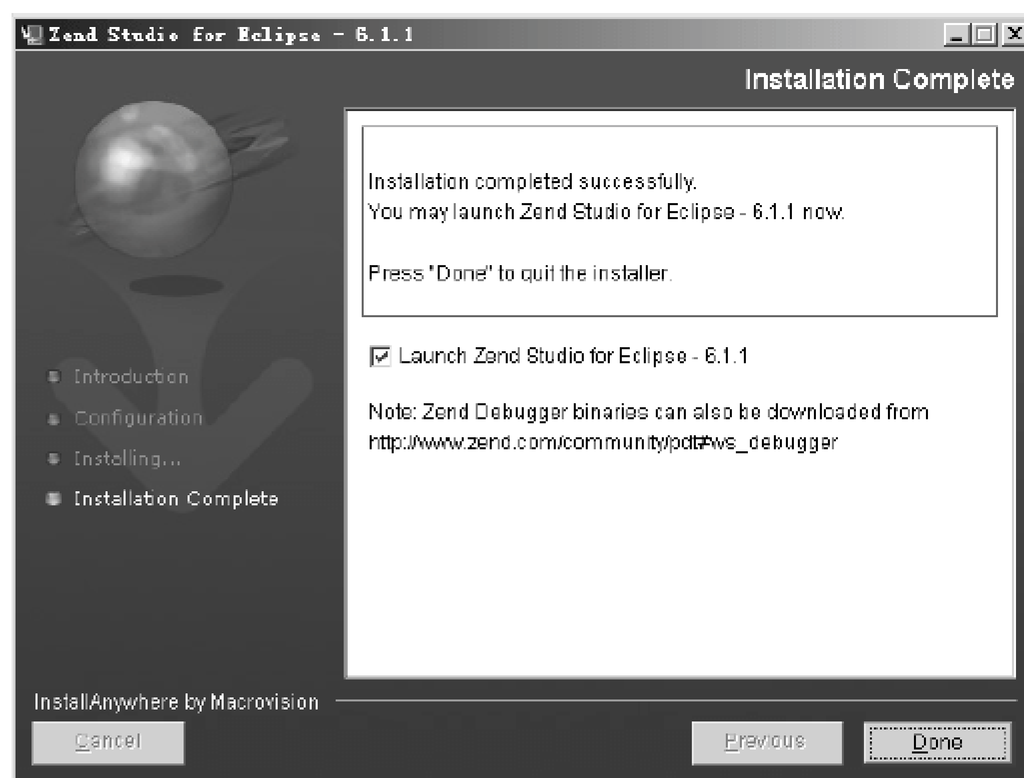


图 1.123 安装成功

这就是 Zend Studio for Eclipse 的安装，过程非常简单，只是等待的时间较长。

秘笈心法

心法领悟 029：选择工作空间。

运行 Zend Studio for Eclipse，可以选择“File”/“Switch_Workspace”命令来选择工作的空间，或者重新创建工作空间，如图 1.124 所示。

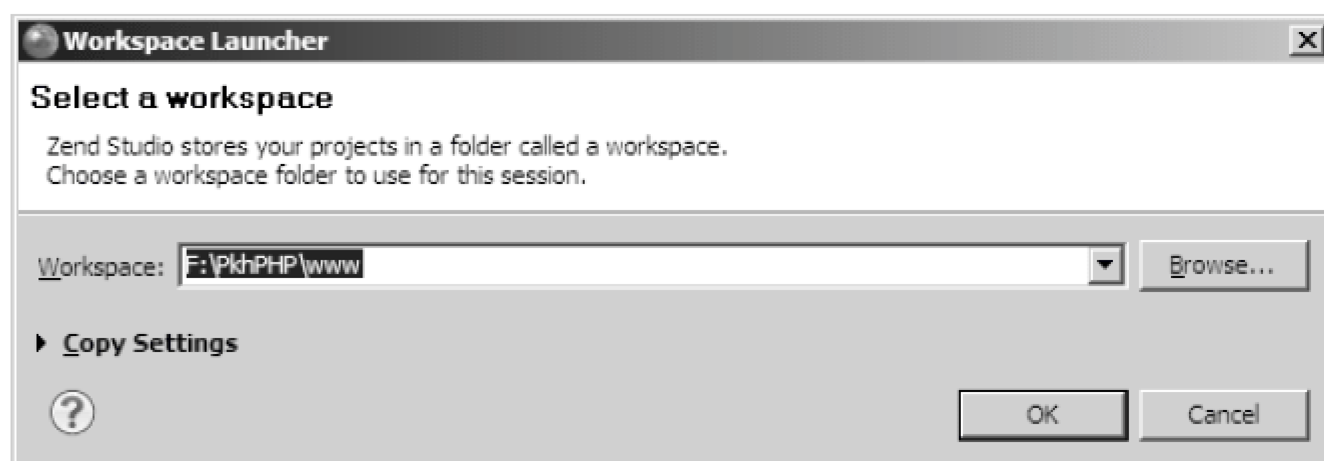


图 1.124 创建工作空间

实例 030 Zend Studio 创建 PHP 项目

光盘位置：光盘\MR\01\030

中级

趣味指数：★★★★☆

实例说明

本实例介绍如何通过 Zend Studio 创建 PHP 项目。在本项目中创建一个 PHP 文件，通过 phpinfo()函数输出 PHP 的配置信息，其运行结果如图 1.125 所示。

关键技术

通过 Zend Studio 编写 PHP 程序，首先要选择正确的工作空间，如果工作空间选择不正确，那么 PHP 的程序就无法运行。

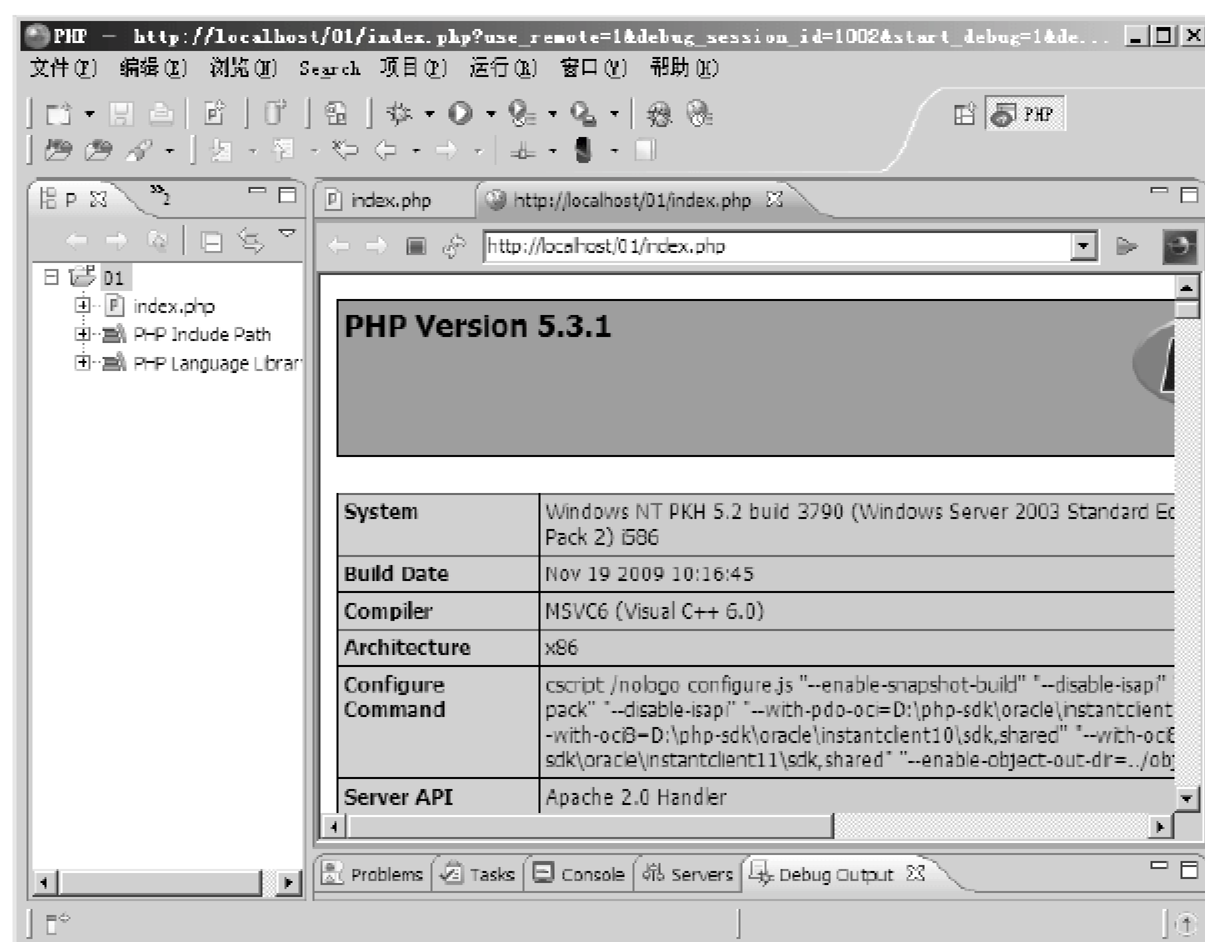


图 1.125　运行程序

设计过程

（1）运行 Zend Studio，首先进入到 Zend Studio 的欢迎页面，如图 1.126 所示。

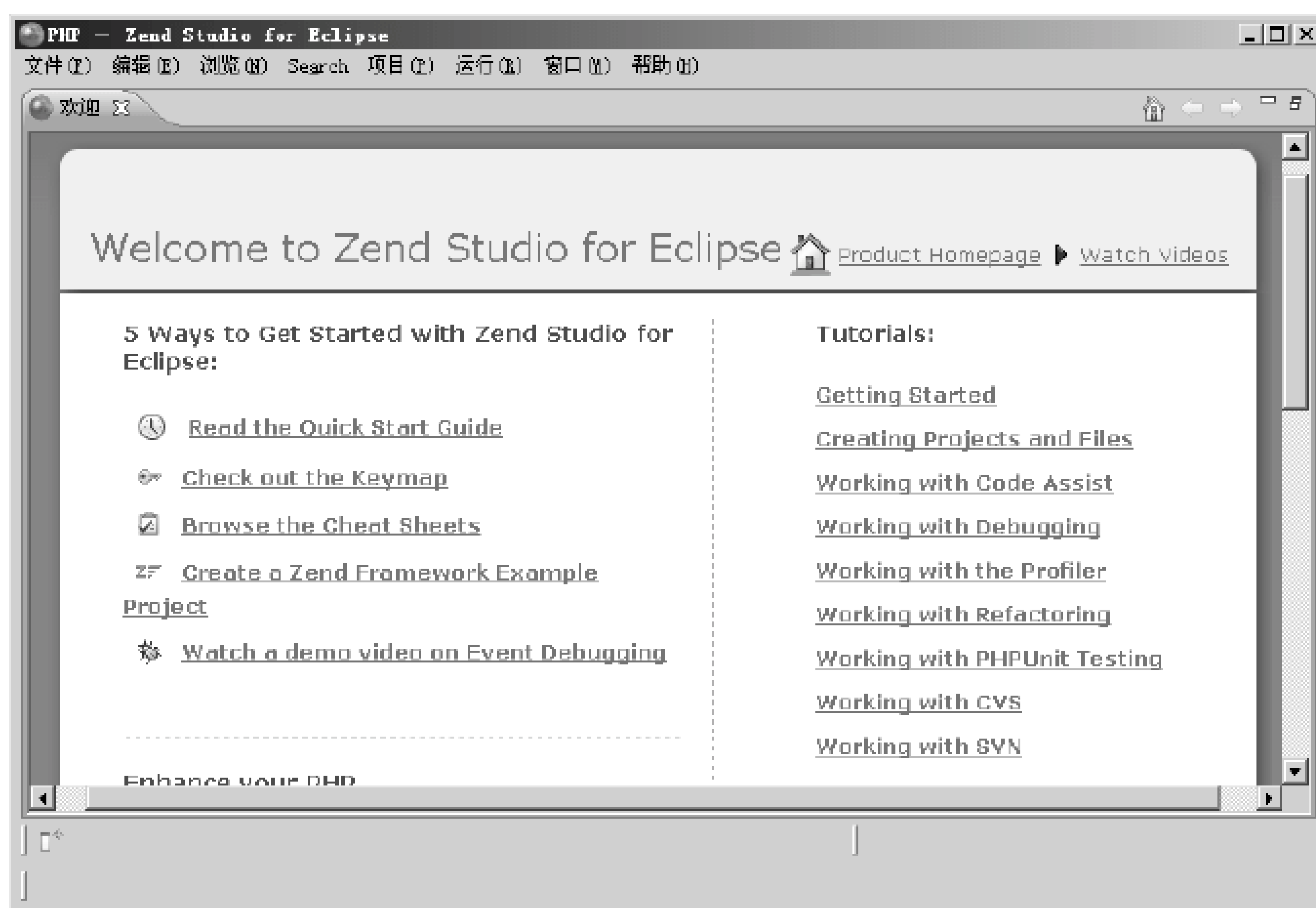


图 1.126　进入到 Zend Studio 的欢迎页面

（2）选择“文件”/“切换工作空间”/“其他”命令，设置工作空间，将项目保存在指定名称的文件夹中，如图 1.127 所示，将 F:\PkhPHP\www 文件夹作为工作空间。

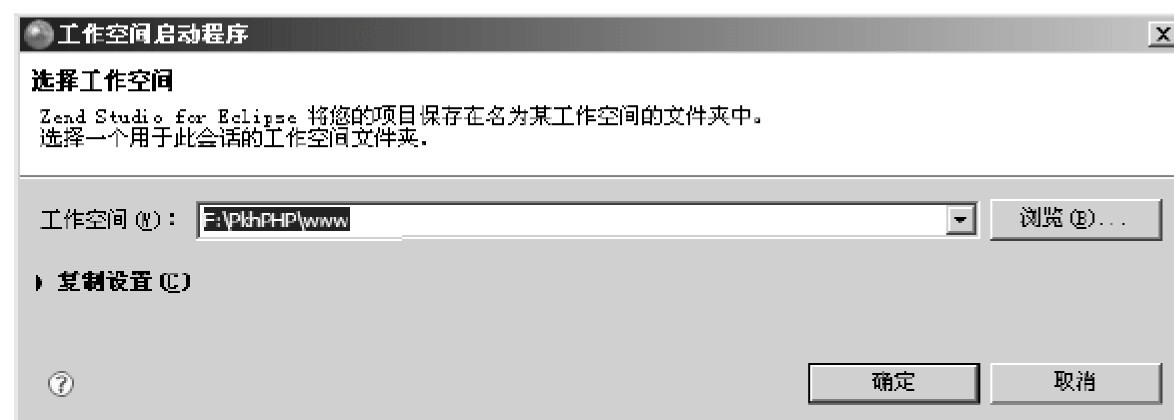


图 1.127　选择工作空间

（3）接下来进入 Zend Studio 的工作台中，如图 1.128 所示。

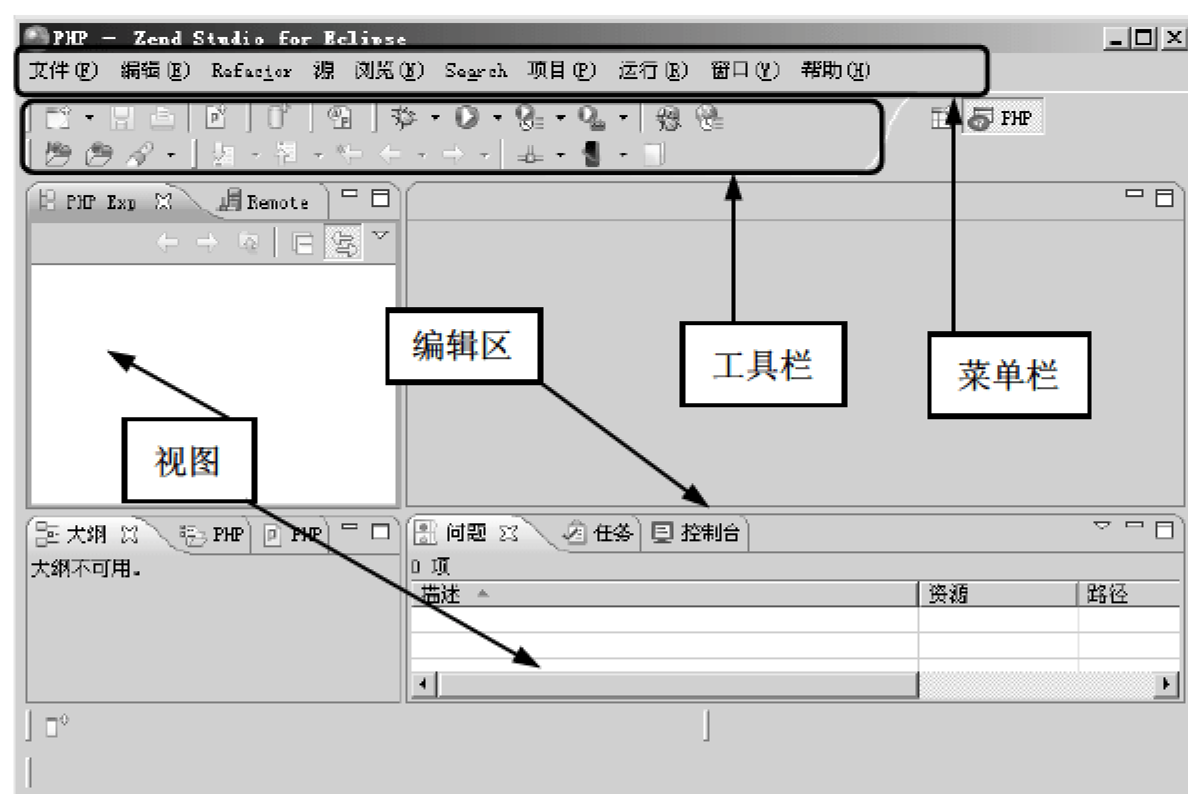


图 1.128　Zend Studio 的工作台

（4）创建一个 PHP 项目。选择“文件”/“新建”/“PHP Project”命令，创建一个 PHP 项目，如图 1.129 所示。

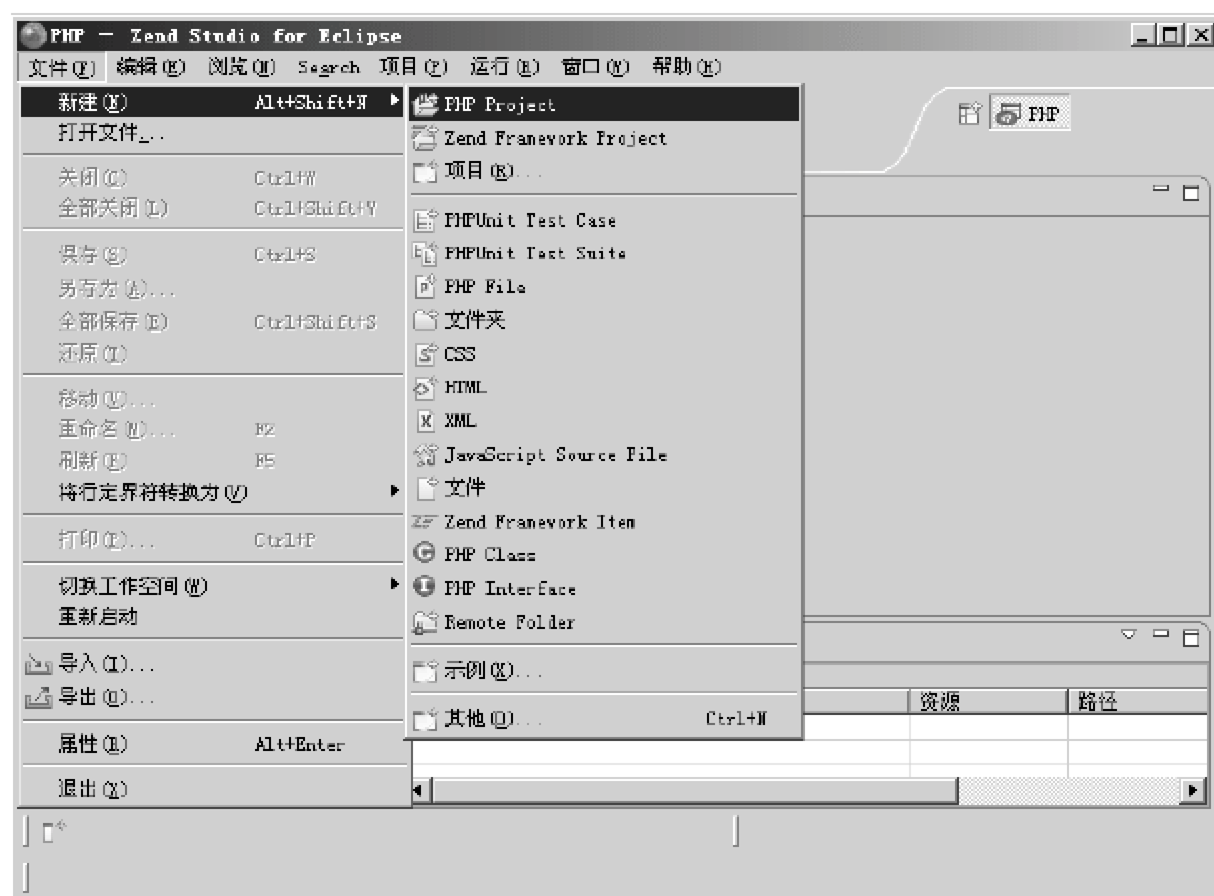


图 1.129　创建 PHP 项目

（5）进入到新项目创建窗口中，在该窗口中设置项目名称为 01；指定项目文件存储位置，这里使用默认值；选择 PHP 的版本，设置是否支持 JavaScript 脚本，如图 1.130 所示。

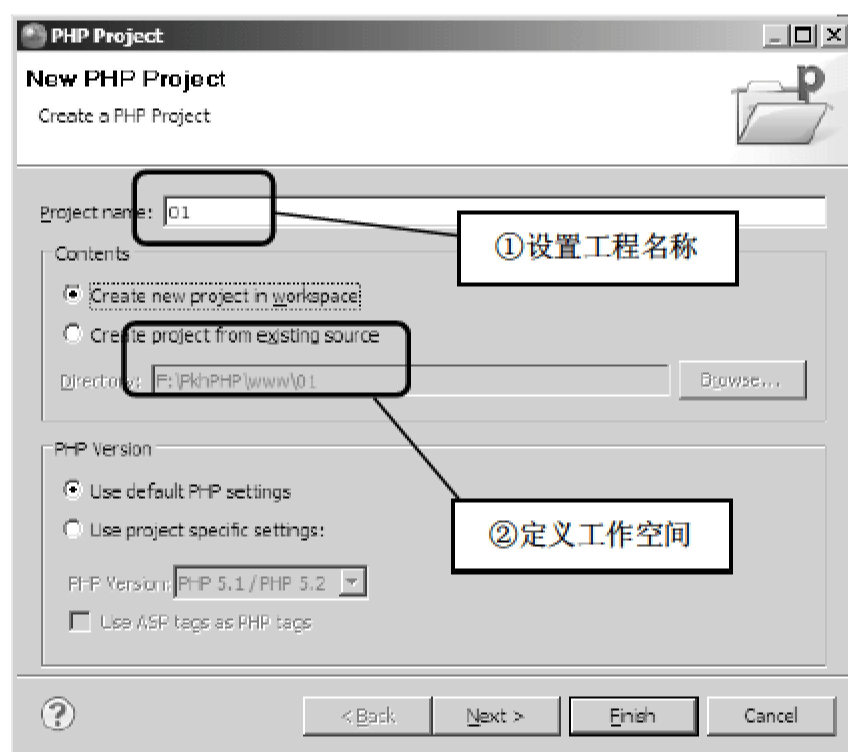


图 1.130　新建工程

（6）在项目 01 中创建 PHP 文件，如图 1.131 所示。

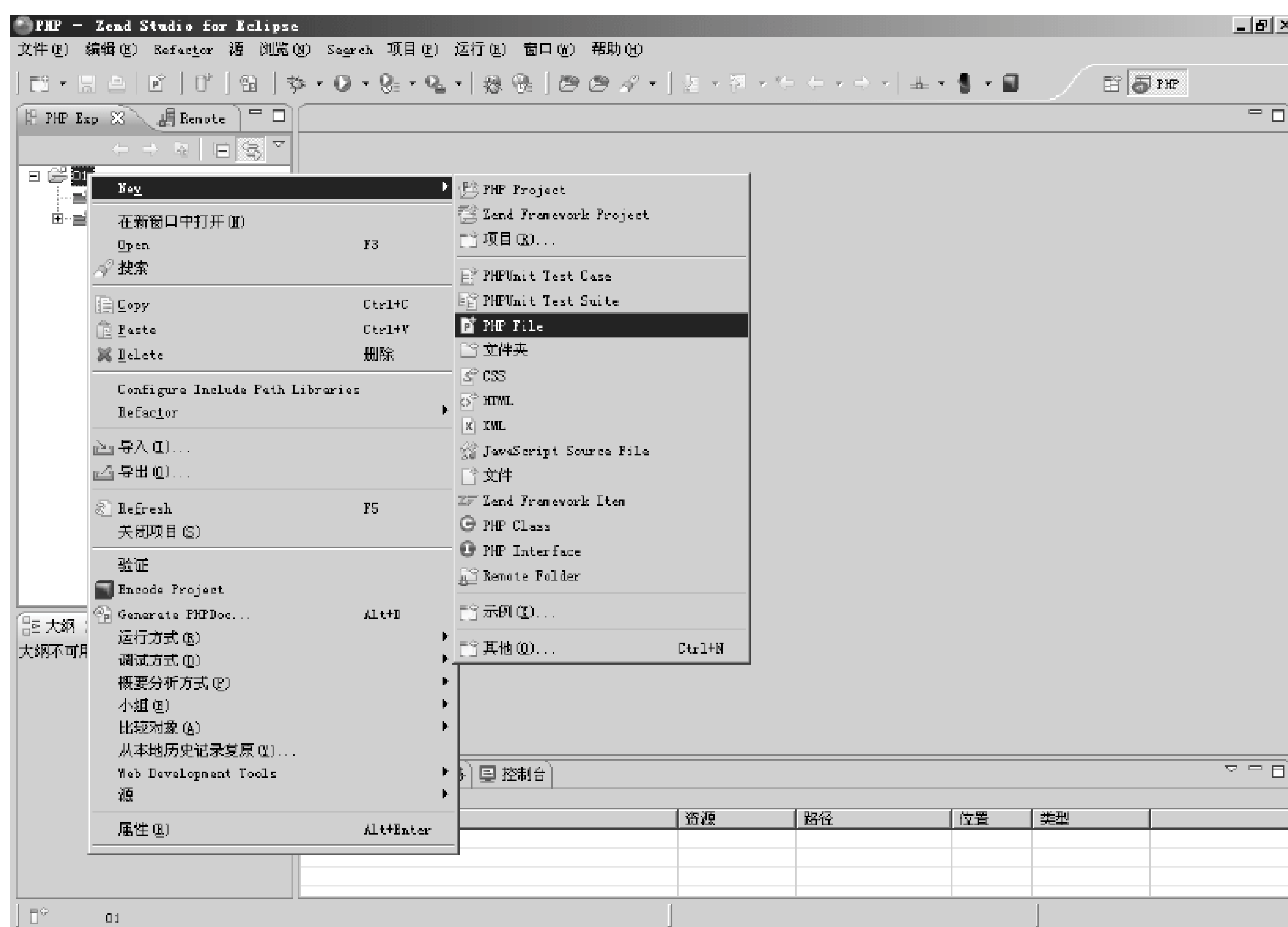


图 1.131　创建 PHP 文件

在图 1.129 所示的操作步骤中，不但可以创建 PHP 文件，还可以创建 html、CSS、JavaScript 和 XML 文件等，具体可以根据程序的实际情况进行创建。

（7）进入到文件创建窗口，定义文件名称，最后单击“完成”按钮，如图 1.132 所示。

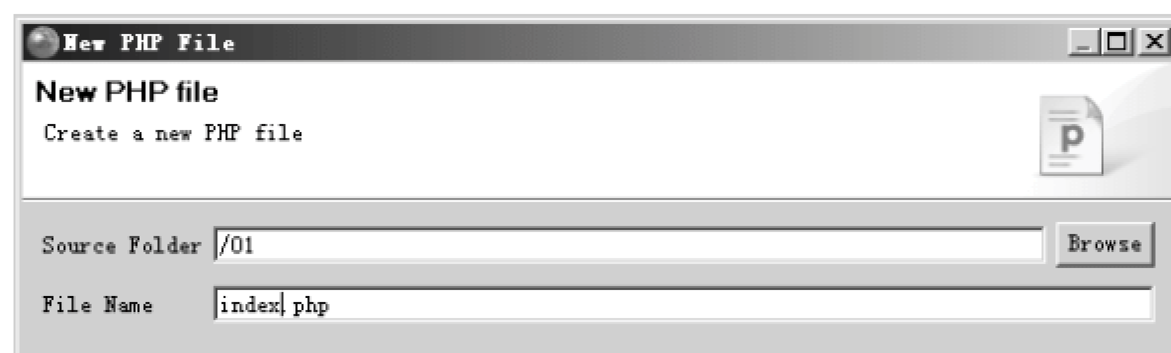


图 1.132　创建 index.php 文件

（8）文件创建成功后，就可以在编辑区中编辑文件的内容，如图 1.133 所示。

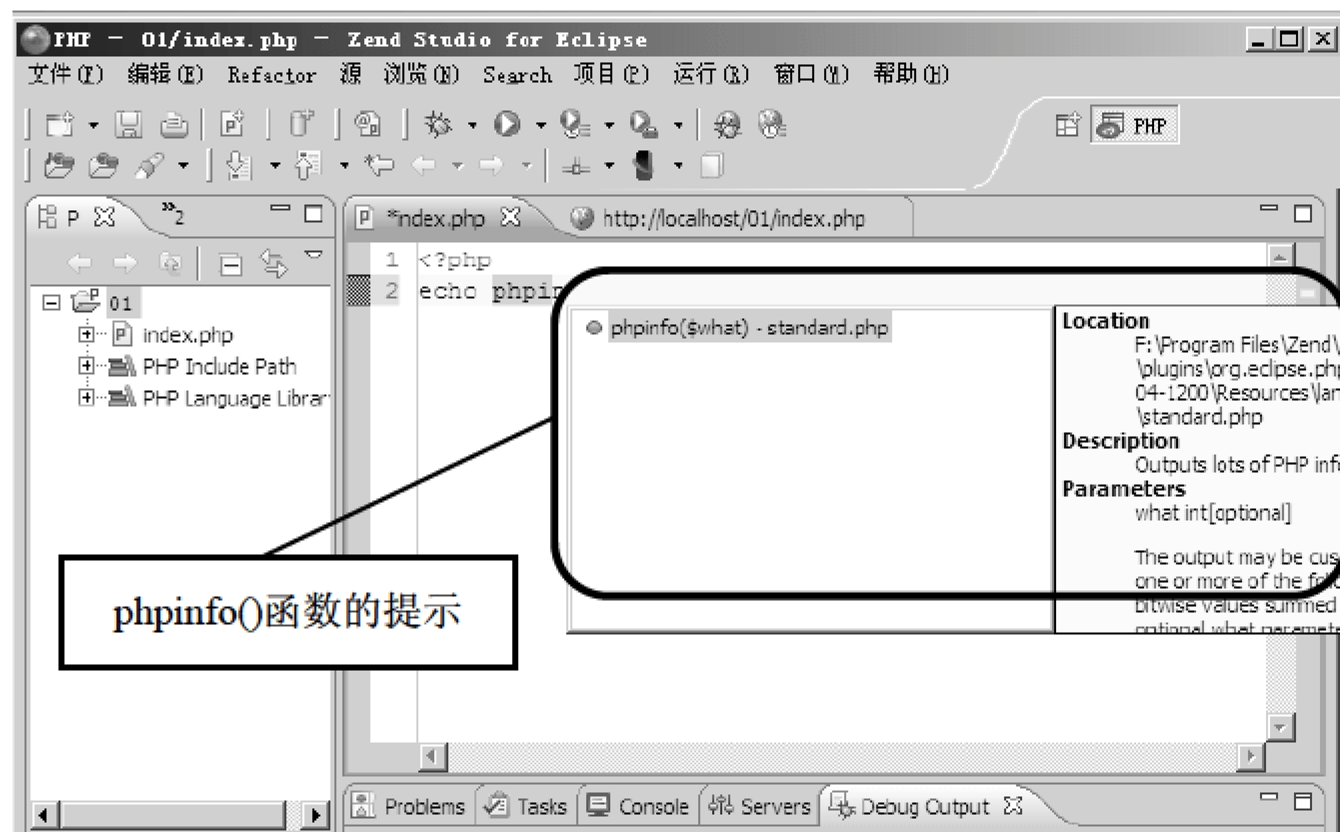


图 1.133　编辑 PHP 代码

如图 1.133 所示，Zend Studio for Eclipse 工具最大的优点就是代码提示功能，它可以在用户输入部分 PHP 脚本语言时，给出相关 PHP 脚本语言的提示供选择。这里应用 phpinfo()函数，获取 PHP 的相关配置信息。

（9）编辑完成后，保存该文件，单击按钮，运行程序，运行结果可以参看实例说明。

秘笈心法

心法领悟 030：Zend Studio 对 Zend Framework 框架的支持。

通过 Zend Studio 不但可以创建 PHP 项目，而且可以直接生成 Zend Framework 框架，开发者不必再为搭建 Zend Framework 框架而头疼。

实例 031 Zend Studio 编码格式的转换

光盘位置：光盘\MR\01\031

中级

趣味指数：★★★★☆

实例说明

在 Zend Studio 中同样可以控制创建项目、文件的编码格式。至于编码格式选择的重要性这里就不再赘述，直接介绍如何在 Zend Studio 中设置项目、文件的编码格式。如果通过 Zend Studio 打开一个编码格式不匹配的文件，那么输出结果将如图 1.134 所示。



图 1.134 编码格式不正确导致乱码

关键技术

在 Zend Studio 中设置项目、文件的编码格式，关键是从根本着手，也就是说要控制一个项目中文件的编码格式，只要项目本身的编码格式确立，那么创建的文件的格式自然是与项目相同的。

设计过程

（1）在 Zend Studio 中要设置项目的编码格式，首先就要创建一个项目。有关项目创建的具体步骤请参看实例 030，这里不再赘述。

（2）项目创建成功后，选中创建的项目，单击右键，在弹出的菜单中选择“Properties”命令，将弹出如图 1.135 所示的对话框。

（3）在图 1.135 所示的对话框中，选中“Other”单选按钮，在其下拉列表中选择指定的编码格式，最后单击 OK 按钮，项目编码格式即设置完成。

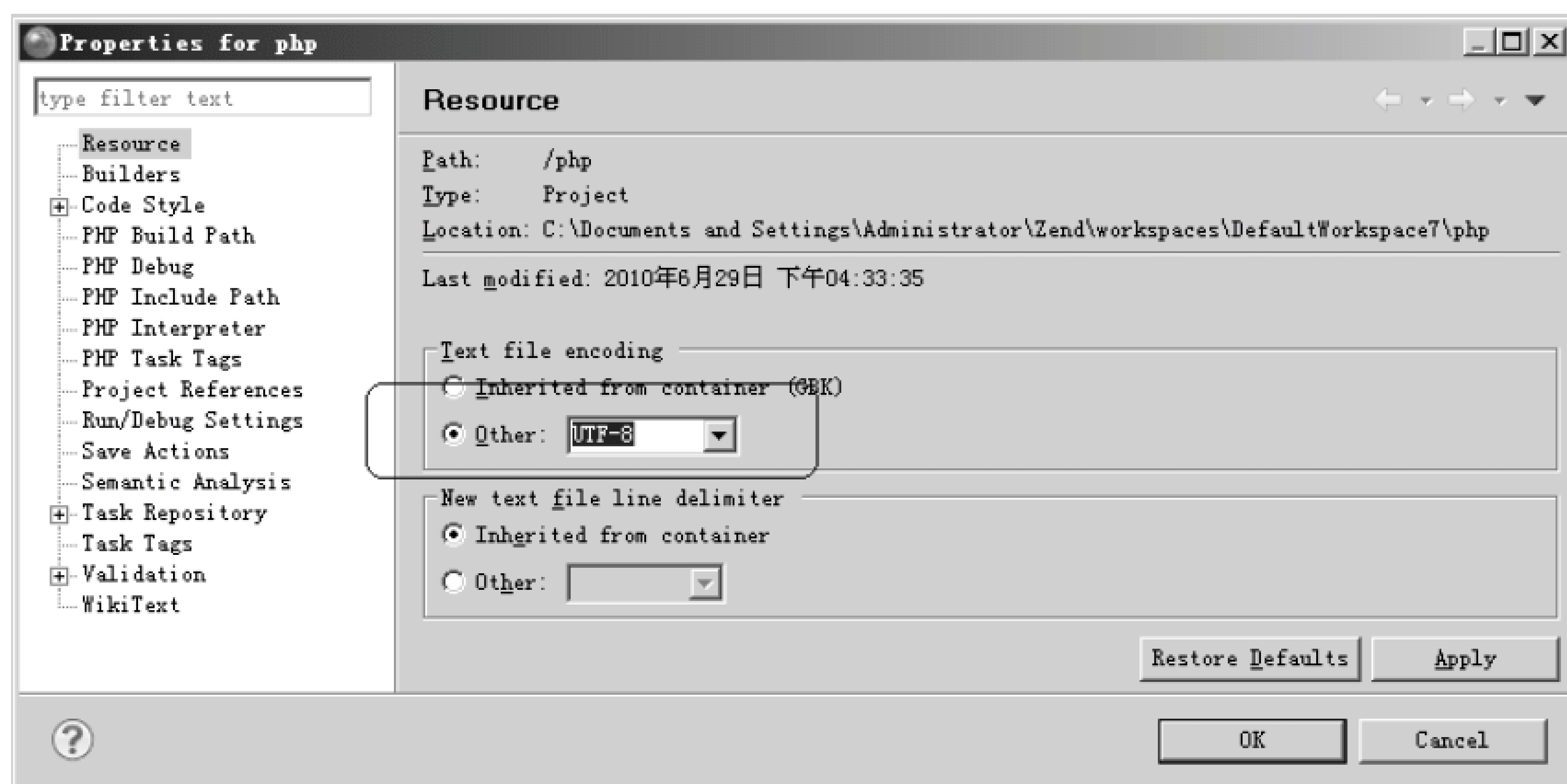


图 1.135　设置项目的编码格式

秘笈心法

心法领悟 031：在 Zend Studio 中修改编码格式。

在实例说明中我们曾看到一个编码格式不匹配的文件打开的效果，那么如何对这个文件的编码格式进行修改呢？

首先，打开编码格式不匹配的文件。然后，单击 Edit 按钮，在弹出的菜单中选择“Set Encoding”命令，将弹出如图 1.136 所示的对话框，在这个对话框中选中“Ohter”单选按钮，设置这个文件的编码格式。最后，单击 OK 按钮。

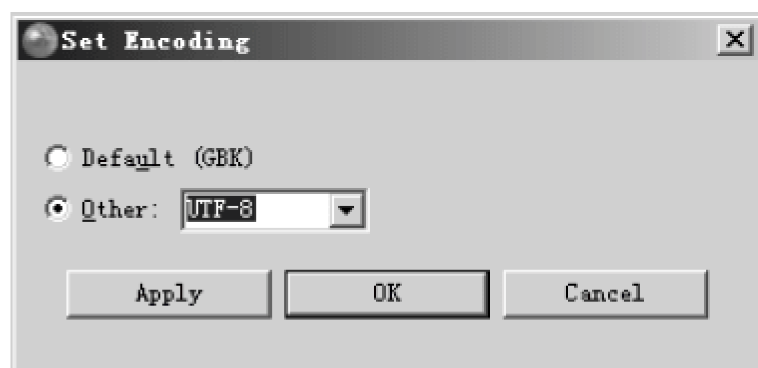


图 1.136　设置文件的编码格式

实例 032	Zend Studio 中快捷键的运用 光盘位置：光盘\MR\01\032	中级 趣味指数：★★★★☆

实例说明

在 Zend Studio 中，通过快捷键可以完成很多操作，快捷键的运用可以为开发者节省很多时间，而且可以简化操作的流程。在本实例中将介绍如何运用 Zend Studio 中的快捷键创建 PHP 项目。

关键技术

要应用 Zend Studio 中的快捷键，必须了解快捷键代表的意义。在 Zend Studio 中，单击 Help 按钮，选择“Key Assist…”命令将弹出 Zend Studio 中的快捷键列表，或者也可以直接按 Ctrl+Shift+L 键，同样可以弹出快捷键列表。在这个列表中可以查看到所有快捷键代表的意义，如图 1.137 所示。

%UpdateToRevision.label	Ctrl+Alt+D
ActionDefinition.renameElement.name	Alt+Shift+V
Activate Editor	F12
Activate Task	Ctrl+F9
Add Block Comment	Ctrl+Shift+/
Add Javadoc Comment	Alt+Shift+J
Add to svn:ignore...	Ctrl+Alt+I
All Instances	Ctrl+Shift+N
Backward History	Alt+Left
Build All	Ctrl+B
Change Method Signature	Alt+Shift+C
Close	Ctrl+W
Close All	Ctrl+Shift+W
Collapse	Ctrl+Numpad_Subtract
Collapse All	Ctrl+Shift+Numpad_Divide
Commit...	Ctrl+Alt+C
Content Assist	Alt+/
Context Information	Alt+?
Copy	Ctrl+C
Copy Lines	Ctrl+Alt+Down
Create Patch...	Ctrl+Alt+P
Cut	Ctrl+X
Deactivate Task	Ctrl+Shift+F9
Debug	F11
Debug Ant Build	Alt+Shift+D, Q

Press "Ctrl+Shift+L" to open the preference page.

图 1.137　Zend Studio 中的快捷键列表

设计过程

（1）在 Zend Studio 中创建一个 PHP 项目。可以直接使用快捷键 Alt+Shift+N，将弹出一个列表，选择“PHP Project”命令，创建一个 PHP 项目，如图 1.138 所示。

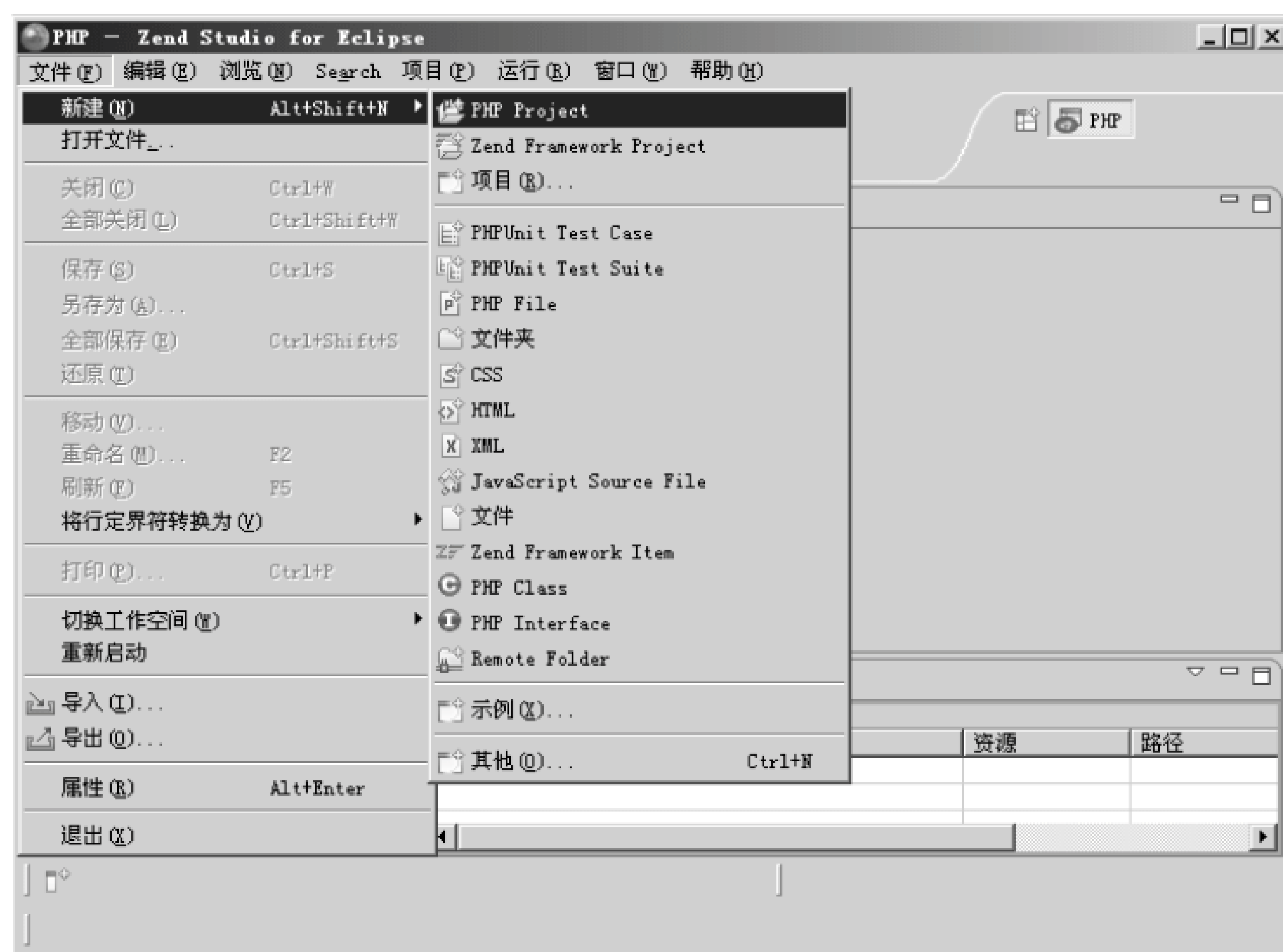


图 1.138　创建 PHP 项目

（2）进入到新项目创建窗口中，在该窗口中设置项目名称为 01；指定项目文件存储位置，这里使用默认值；选择 PHP 的版本，设置是否支持 JavaScript 脚本，如图 1.139 所示。

（3）在项目 01 中创建 PHP 文件。编写一个简单实例，同样可以使用快捷键 Alt+Shift+N，只是要在列表中选择“PHP File”命令，如图 1.140 所示。

（4）进入到文件创建窗口，定义文件名称，最后单击“完成”按钮，完成文件的创建，如图 1.141 所示。

（5）文件创建成功后，就可以在编辑区中编辑文件的内容。在编辑区中可以通过快捷键 Ctrl+Shift+F 对代码进行格式化，如图 1.142 所示。

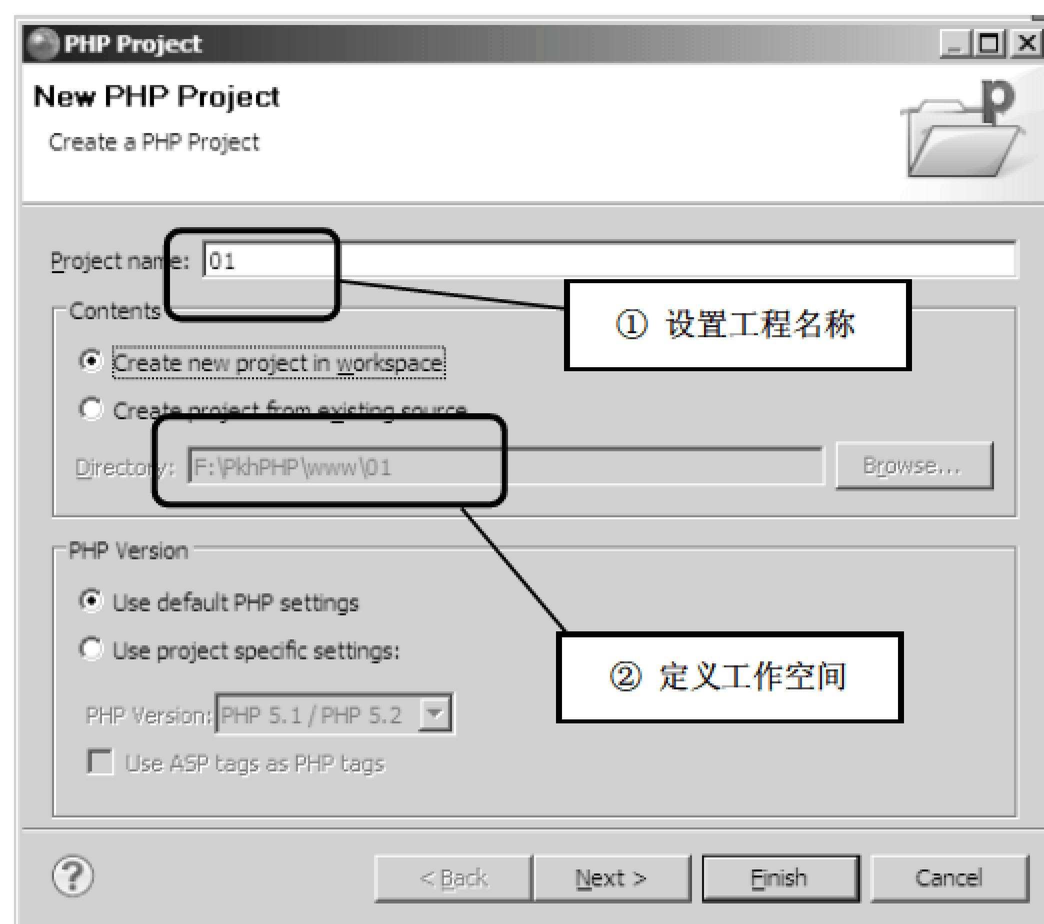


图 1.139　新建工程

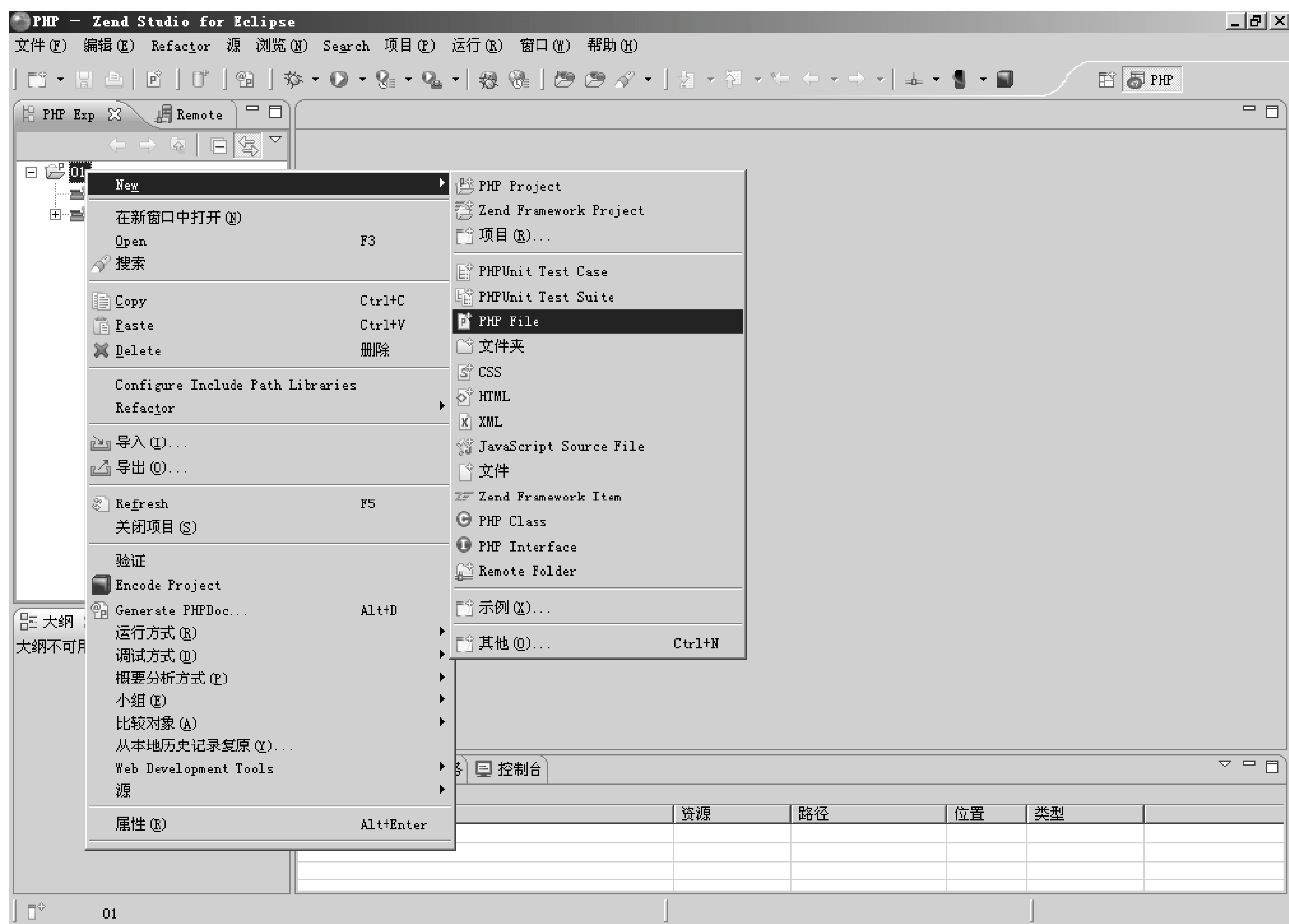


图 1.140　创建 PHP 文件

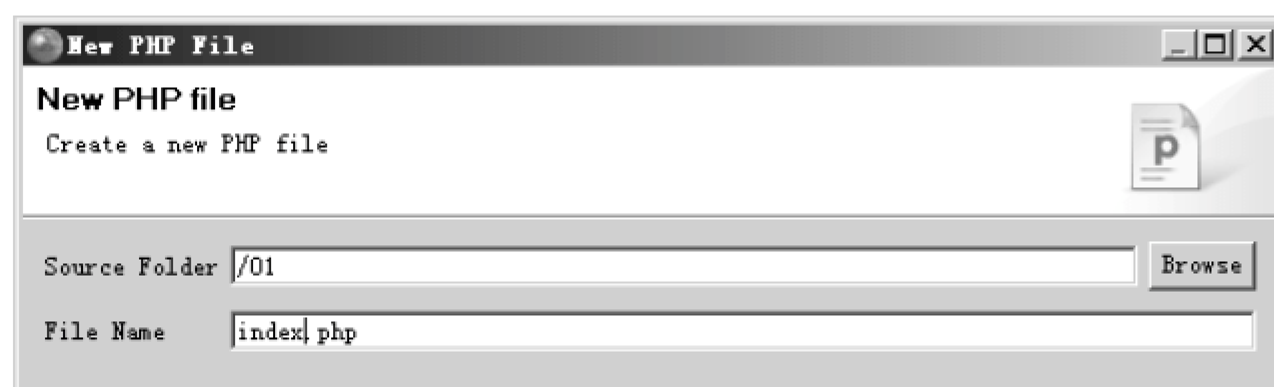


图 1.141　创建 index.php 文件

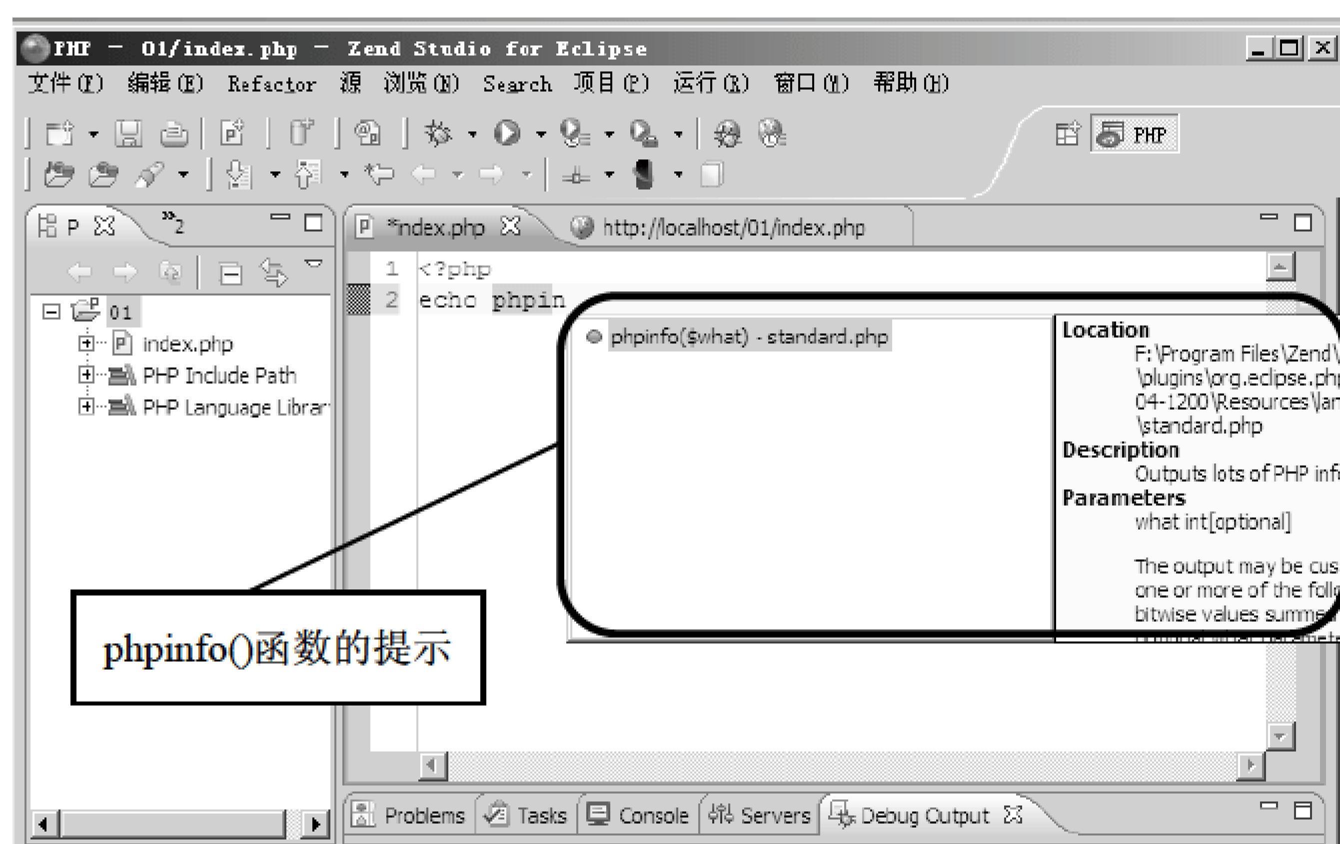


图 1.142　编辑 PHP 代码

（6）编辑完成后，保存该文件，单击按钮，运行程序，运行结果如图 1.142 所示。

秘笈心法

心法领悟 032：在 Zend Studio 中使用快捷键。

在 Zend Studio 中，对于快捷键的使用不必机械的去记忆，只要了解一些常用的就可以。对于初学者来说，通过鼠标进行操作比使用快捷键更好，更有助于对工具功能的理解和学习。

实例 033　Zend Studio 中部署 Apache 服务器

光盘位置：光盘\MR\01\033

中级

趣味指数：★★★★☆

实例说明

在运行 PHP 程序的过程中，需要将 PHP 程序复制到 Apache 服务器指定的根目录下，然后在 IE 浏览器的地址栏中输入 http://127.0.0.1/或者 http://localhost/来运行 PHP 程序。而在 Zend Studio 中可以直接部署 Apache 服务器，部署完成后直接单击按钮，运行 PHP 程序，其运行结果如图 1.143 所示。

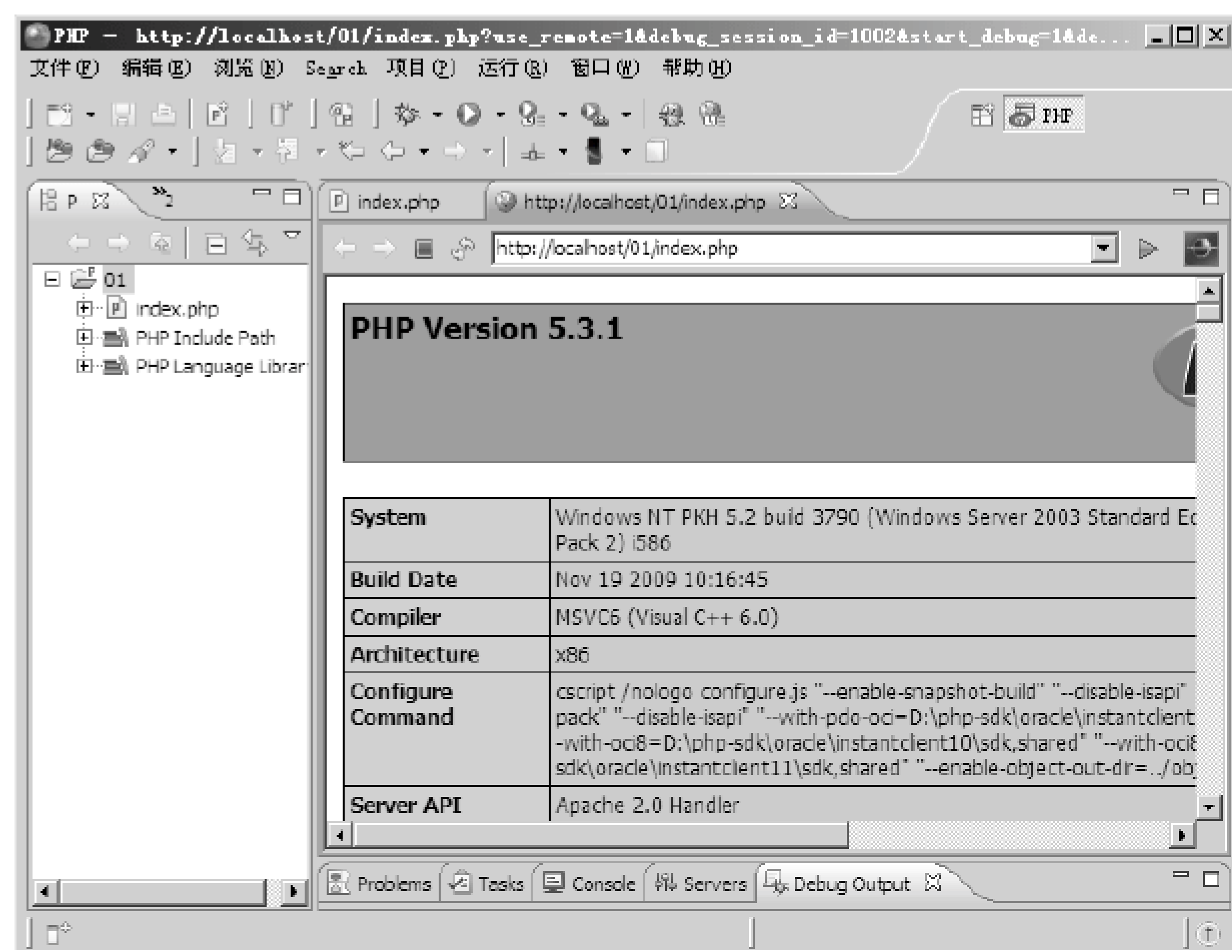


图 1.143　运行程序

关键技术

在 Zend Studio 中部署 Apache 服务器，关键是指定正确的访问路径，如图 1.144 所示。

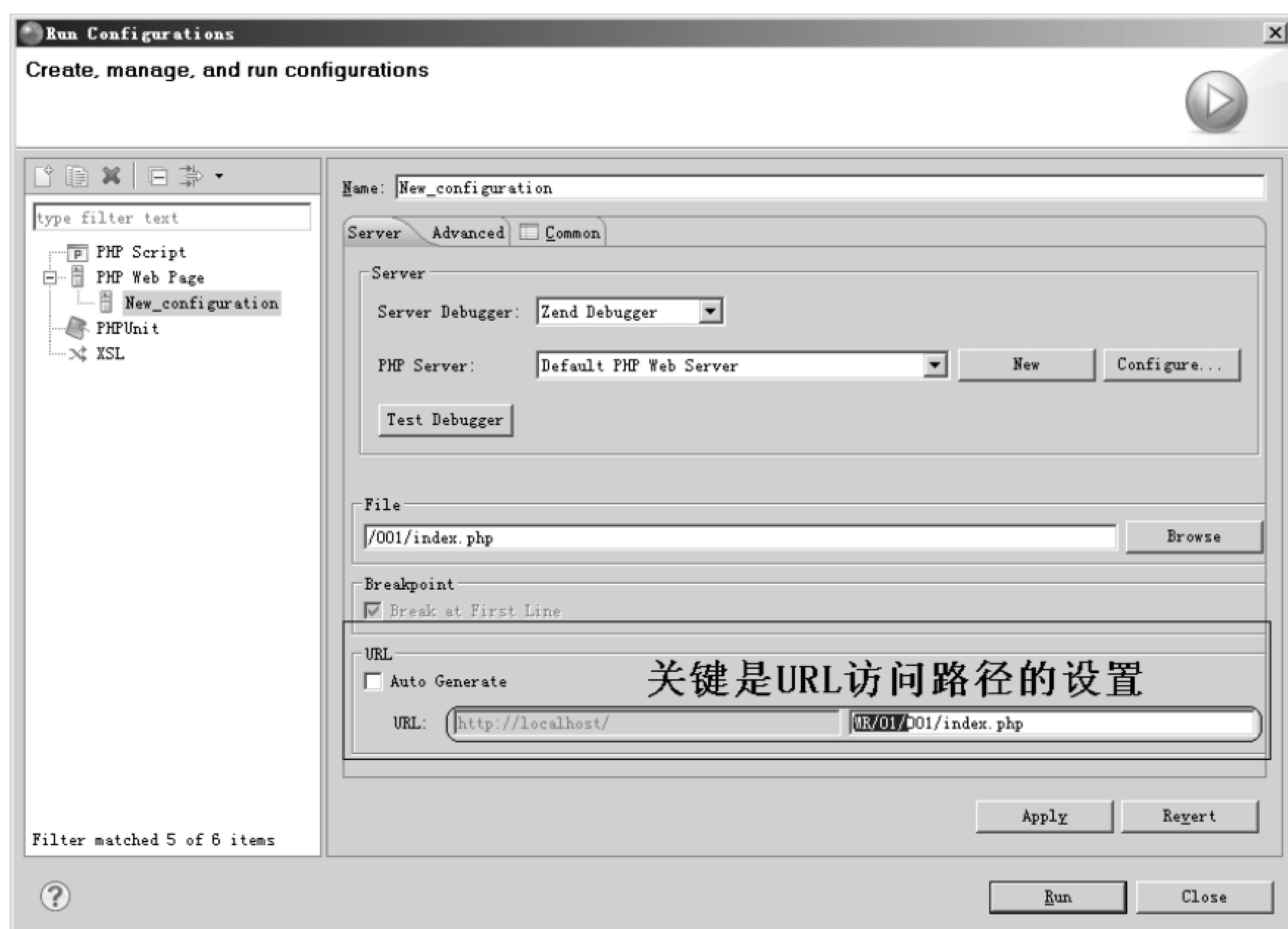


图 1.144　部署 Apache 服务器的访问路径

如果在图 1.144 所示的界面中，没有指定 URL 中访问文件的具体位置 MR/01/001，那么程序就不能正常运行，将出现如图 1.145 所示的效果。

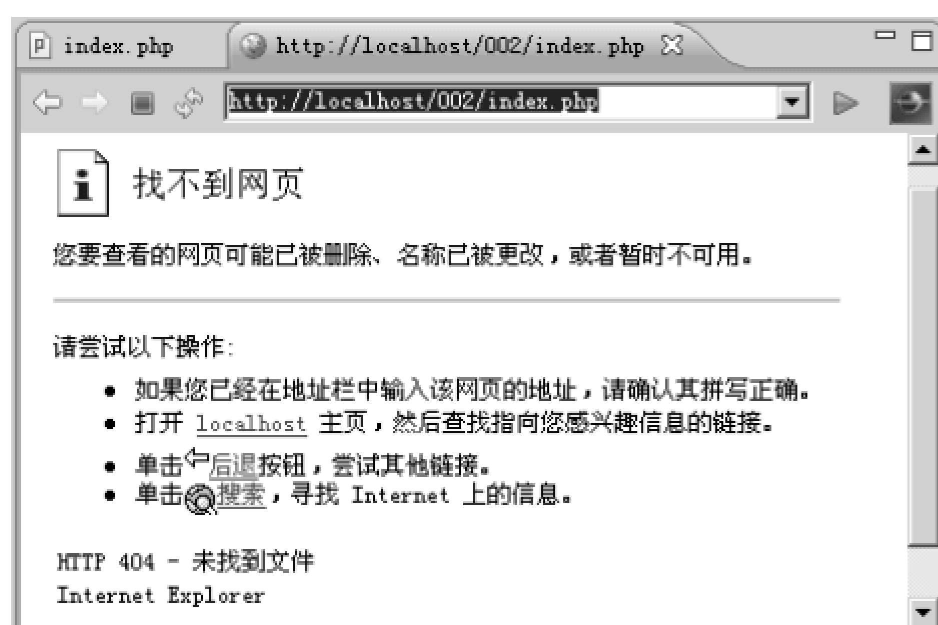


图 1.145　错误的设置访问路径

设计过程

在 Zend Studio 中部署 Apache 服务器的操作步骤如下：

（1）创建一个 PHP 项目，这里不再赘述。

（2）单击工具栏中按钮的三角标，将弹出一个列表框，选择 Run Configurations 选项，在弹出的对话框中完成对 Apache 服务器的部署，包括文件的指定，如图 1.146 所示。URL 访问路径的设置，如图 1.147 所示。



图 1.146　设置访问文件

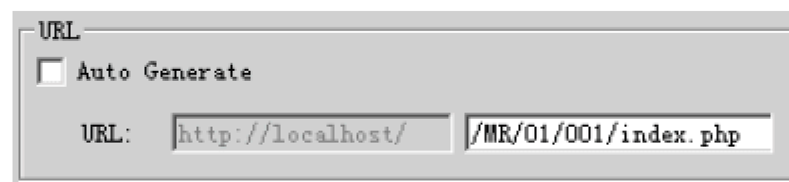


图 1.147　设置 URL 访问路径

（3）设置完成后单击 Run 按钮，即可浏览 PHP 项目中的内容。

秘笈心法

心法领悟 033：在 Zend Studio 中修改快捷键的设置。

单击 Window 按钮，选择“Preferences”命令，将弹出如图 1.148 所示的对话框，可以在此对话框中完成对快捷键的修改操作。

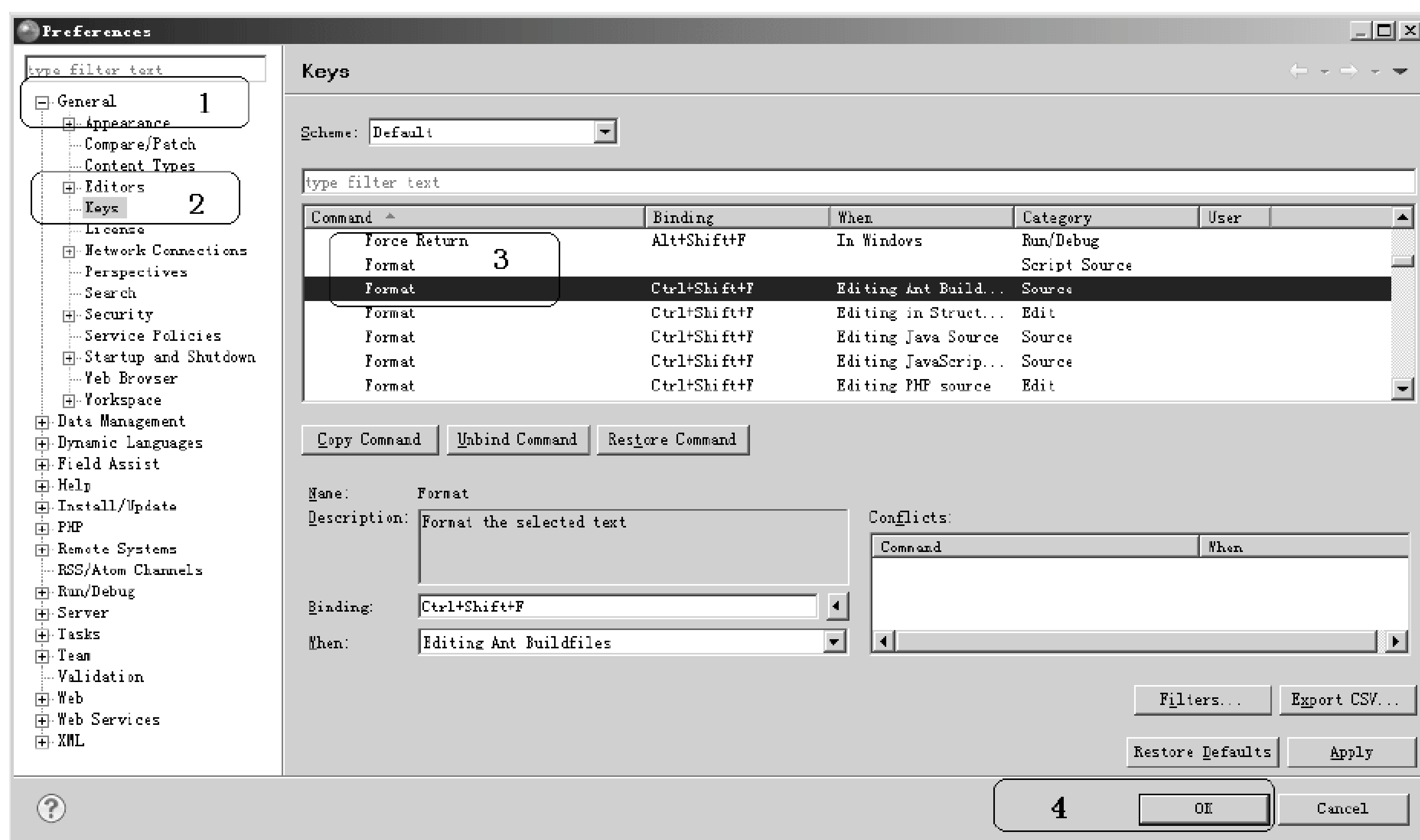


图 1.148　Zend Studio 中快捷键的修改步骤

第 2 章

PHP 基础

- 基本语法
- 运算符
- 包含语句
- 条件语句
- 循环控制
- 跳转语句
- 自定义函数
- 字符串
- 正则表达式
- 数组
- 日期和时间

2.1 基 本 语 法

也许你是一个 PHP 初学者，当你向有经验的 PHP 高级程序员请教如何学好 PHP 这门语言时，想必他会问你有没有其他编程语言的基础，如果有的话，就能够很快入门，学起来也很轻松。

由此可见，掌握一门语言是多么的重要。可以说，扎实的语言基础，是培养和提高程序员编程能力的基本途径，语言基础也是学习 PHP 过程中非常重要的一个环节，它在很大程度上决定了程序员 PHP 学习的进展效果。

另外，程序员的创造力极强，可以完全按照自己的设计目标和思路组织代码，用与其他程序员完全不同的设计方法实现完全相同的功能。但是，编程同时也是一项十分严谨的工作，因为语法是严格限定的，代码必须严格遵守语法要求进行书写，任何不符合规范的代码都不能被计算机执行，哪怕是程序员写错一个字母，或者少写一个结束符都会使程序出错。

所以，程序设计语言的语法必须得到必要的重视。语法是 PHP 语言的规则，打好这个基础，循序渐进地学习，实际运用时才会得心应手。

实例 034	在页面中打印 PHP 的配置信息 光盘位置：光盘\MR\02\034	高级 趣味指数：★★★★

实例说明

本实例应用 PHP 的标记通过 echo 语句和 phpinfo()函数，向用户展示 PHP 的配置信息，包括配置文件所在的目录以及一些相关扩展库的版本、作者信息等。运行结果如图 2.1 所示。

PHP Version 5.2.6

php

System	Windows NT LM 5.2 build 3790
Build Date	May 2 2008 18:01:20
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared" "--with-extra-includes=C:\Program Files (x86)\Microsoft SDK\Include;C:\PROGRA~2\MICROS~2\VC98\ATL\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\INCLUDE;C:\PROGRA~2\MICROS~2\VC98\MFC\INCLUDE" "--with-extra-libs=C:\Program Files (x86)\Microsoft SDK\Lib;C:\PROGRA~2\MICROS~2\VC98\LIB;C:\PROGRA~2\MICROS~2\VC98\MFC\LIB"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS
Loaded Configuration File	C:\WINDOWS\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, data, http, ftp, compress.zlib
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, zlib.*

图 2.1　PHP 配置信息

关键技术

本实例的关键点是应用 PHP 的输出语句 echo，输出 phpinfo()函数的返回值。phpinfo()函数的代码如下：

```
int phpinfo ( [int what]);
```

该函数将输出 PHP 的配置信息。

设计过程

创建 index.php 文件，写 PHP 标记，通过 echo 语句输出 phpinfo()函数的返回值，其代码如下：

```
<?php
echo phpinfo();                    //输出 PHP 配置信息
?>
```

秘笈心法

心法领悟 034：phpinfo()函数的具体应用。

在实际的 PHP 项目的开发过程中，用户加载的任何一个组件，在 phpinfo()函数的输出页面中都会显示是否已经获得支持，所以此函数的应用对于一个初学者来说是十分重要的。

实例 035	在页面中打印服务器时间 光盘位置：光盘\MR\02\035	高级 趣味指数：★★★★

实例说明

在用 PHP 编写的网站中，经常需要获取当前时间。例如用户在什么时间登录的网站、黑客攻击网站是在什么时间等。本实例应用 PHP 标记，通过 echo 语句和 date()函数来实现获取服务器时间，并进行打印。运行结果如图 2.2 所示。



图 2.2　系统当前时间

关键技术

本实例主要是通过 date()函数获取系统的格林威治时间，其语法及说明如下：

```
date(string format,int timestamp);
```

date()：参数 format 指定日期和时间输出的格式。有关参数 format 指定的格式如表 2.1 所示。参数 timestamp 为可选参数，指定时间戳，如果没有指定时间戳，则使用本地时间 time()。

表 2.1　参数 format 的格式化选项

参　数	说　明
a	小写的上午和下午值，返回值 am 或 pm
A	大写的上午和下午值，返回值 AM 或 PM
B	Swatch Internet 标准时间，返回值 000~999
d	月份中的第几天，有前导零的 2 位数字，返回值 01~31
D	星期中的第几天，文本格式，3 个字母，返回值 Mon~Sun

续表

参　数	说　明
F	月份，完整的文本格式，返回值 January 到 December
g	小时，12 小时格式，没有前导零，返回值 1~12
G	小时，24 小时格式，没有前导零，返回值 0~23
i	有前导零的分钟数，返回值 00~59
I	判断是否为夏令时，如果是夏令时返回值为 1，否则为 0
j	月份中的第几天，没有前导零，返回值 1~31
l	星期数，完整的文本格式，返回值 Sunday~Saturday
L	判断是否为闰年，如果是闰年返回值为 1，否则为 0
m	数字表示的月份，有前导零，返回值 01~12
M	3 个字母缩写表示的月份，返回值 Jan~Dec
n	数字表示的月份，没有前导零，返回值 1~12
o	与格林威治时间相差的小时数，如 0200
r	RFC 822 格式的日期，如 Thu，21 Dec 2000 16：01：07 +0200
s	秒数，有前导零，返回值 00~59
S	每月天数后面的英文后缀，两个字符，如 st、nd、rd 或者 th，可以和 j 一起使用
t	指定月份所应有的天数
T	本机所在的时区
U	从 UNIX 纪元（January 1 1970 00:00:00 GMT）开始至今的秒数
w	星期中的第几天，数字表示，返回值为 0~6
W	ISO-8601 格式年份中的第几周，每周从星期一开始
y	2 位数字表示的年份，返回值如 88~08
Y	4 位数字完整表示的年份，返回值如 1998、2008
z	年份中的第几天，返回值 0~366
Z	时差偏移量的秒数。UTC 西边的时区偏移量总是负的，UTC 东边的时区偏移量总是正的，返回值：−43200~43200

设计过程

创建 index.php 文件，通过 echo 语句输出 date()函数的返回值，其代码如下：

```
<?php
echo date('Y-m-s H:i:s');                    //输出当前时间
?>
```

秘笈心法

心法领悟 035：对系统时间的设置。

看了上面的实例，读者也许会感到奇怪，为什么系统当前的时间比正常的时间少了 8 个小时？原因是系统输出的当前时间是系统默认的格林威治时间，而不是我们所使用的东八区的北京时间。

想要更改为东八区的时间，有两种方法：

（1）修改 php.ini 文件中的设置，找到[date]下的“;date.timezone = ”选项，将该项修改为“date.timezone =Asia/Hong_Kong”，然后重启 Apache 服务器。

（2）在程序中，通过在使用日期格式之前添加“date_default_timezone_set(timezone);”函数来完成对时区的设置。参数 timezone 为 PHP 可识别的时区名称，如果 PHP 无法识别时区名称，则系统采用 UTC 时区。在 PHP 手册中提供了各时区的名称列表，其中北京时间可以设置时区为 PRC（中华人民共和国）、Asia/Chongqing（重庆）、Asia/Shanghai（上海）、Asia/Urumqi（乌鲁木齐），这几个时间是等效的。

说明： 系统默认的格林威治标准时间（Greenwich Mean Time，GMT）与北京时间相差 8 个小时，在国际无线电通信领域，使用一个统一的时间，称为通用协调时间（Universal Time Coordinated，UTC）。UTC 与格林威治标准时间相同。

设置完成后，时间显示如图 2.3 所示。



图 2.3　系统设置后的当前时间

实例 036　在页面中打印当前执行的 PHP 文件名

光盘位置：光盘\MR\02\036

高级

趣味指数：★★★★

实例说明

很多时候用户需要编写包含文件路径及文件名称的代码，如果通过目录查找，未免有些麻烦。这时用户可以使用__FILE__预定义函数。本实例通过__FILE__预定义常量获取目标文件的路径及文件名称并在网页上进行打印，运行结果如图 2.4 所示。



图 2.4　输出当前文件路径和名称

关键技术

系统预定义常量和用户自定义常量在使用上没有差别。大多数预定义常量的执行结果都是服务器的相关信息（版本号、路径、错误参数等），所以程序员很少将此函数用于网站前台的开发，如果被别有用心的人知道了这些信息，会严重威胁服务器的安全。使用此函数的语法说明如下：

__FILE__预定义常量：文件的完整路径和文件名。如果用在包含文件中，则返回包含文件名。自 PHP 4.0.2 起，__FILE__ 总是包含一个绝对路径，而在此之前的版本有时会包含一个相对路径。

设计过程

创建 index.php 文件，通过 echo 语句输出__FILE__预定义常量，其代码如下：

```
<?php
echo __FILE__;                          //获取文件的完整路径
?>
```

秘笈心法

心法领悟 036：预定义常量的说明。

PHP 中可以使用预定义常量获取 PHP 中的信息，常用的预定义常量如表 2.2 所示。

表 2.2　PHP 的预定义常量

常　量　名	功　　能
__FILE__	默认常量，PHP 程序文件名
__LINE__	默认常量，PHP 程序行数
PHP_VERSION	内建常量，PHP 程序的版本，如 3.0.8_dev
PHP_OS	内建常量，执行 PHP 解析器的操作系统名称，如 Windows
TRUE	该常量是一个真值（true）
FALSE	该常量是一个假值（false）
NULL	一个 null 值
E_ERROR	该常量指到最近的错误处
E_WARNING	该常量指到最近的警告处
E_PARSE	该常量指到解析语法有潜在问题处
E_NOTICE	该常量指到不寻常但不一定是错误处

注意：__FILE__和__LINE__中的“__”是两条下划线，而不是一条“_”。表中以 E_开头的预定义常量是 PHP 的错误调试部分。如果想详细了解，请参考 error_reporting()函数。

实例 037　**区分单引号和双引号**　高级

光盘位置：光盘\MR\02\037　趣味指数：★★★★

实例说明

输出或定义字符串时离不开单引号和双引号的修饰，表面上看它们似乎没有什么不同，实际上是有区别的。本实例应用单引号与双引号操作符向用户说明如何区分单引号和双引号，运行结果如图 2.5 所示。

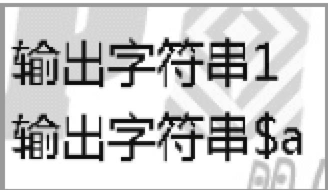


图 2.5　区分单引号和双引号

关键技术

本实例的关键点是输出单引号和双引号修饰的字符串以显示其区别。

使用双引号与单引号最大的区别是，双引号中所包含的变量会自动被替换成实际数值，而在单引号中包含的变量则按普通字符串输出。

设计过程

创建 index.php 文件，通过 echo 语句输出用单、双引号修饰的字符串，其代码如下：

```
<?php
$a = 1;
echo "输出字符串$a"."<br>";
echo '输出字符串$a';
?>
```

秘笈心法

心法领悟 037：转义字符。

通过双引号定义字符串，可以支持很多的转义字符。如表 2.3 所示为双引号支持的转义字符。

表 2.3　双引号支持的转义字符

转 义 字 符	含　义
\n	换行（LF 或 ASCII 字符 0x0A（10））
\r	回车（CR 或 ASCII 字符 0x0D（13））
\t	水平制表符（HT 或 ASCII 字符 0x09（9））
\\	反斜杠
\$	美元符号
\'	单引号
\"	双引号
\[0-7]{1,3}	此正则表达式序列匹配一个用八进制符号表示的字符，如\467
\x[0-9A-Fa-f]{1,2}	此正则表达式序列匹配一个用十六进制符号表示的字符，如\x9f

\n 和\r 在 Windows 系统中没有什么区别，都可以当作回车符，但在 Linux 系统中则是两种效果。在 Linux 中，\n 表示换到下一行，却不会回到行首；而\r 表示光标会回到行首，但仍然在本行。如果读者使用 Linux 操作系统，可以尝试一下这两个转义字符的应用。

技巧： 在定义简单的字符串时，使用单引号是比较合适的处理方式。如果使用双引号，PHP 将花费一些时间来处理字符串的转义和变量的解析。因此，笔者建议在定义字符串时，如果没有特别的要求，应该尽量使用单引号。通过单引号定义的字符串类型变量按照普通字符进行输出。例如，通过单引号定义字符串“'$result”，那么将输出“$result”，而不会输出变量$result 的实际值。

说明：（1）在字符串中表示单引号，需要使用反斜线（\）进行转义。例如，定义一个字符串“It's dogged that does it.”，则需要写出“It\'s dogged that does it.”。

（2）在单引号或者字符串结束位置表示反斜线，同样要用反斜线进行转义，即用两个反斜线（\\）表示。例如，表示字符串“F:\PkhPHP\www”，则要写出“F:\\PkhPHP\\www”。

实例 038　动态输出 JavaScript 代码　高级

光盘位置：光盘\MR\02\038　趣味指数：★★★★

实例说明

JavaScript 语言是一门功能强大的客户端脚本语言，也是一门跨平台语言。PHP 支持使用 JavaScript 编码。本实例通过 PHP5 新型字符串动态输出 JavaScript 代码，运行结果如图 2.6 所示。



图 2.6　动态输出 JavaScript 代码

关键技术

PHP 5.0 新型字符串的使用是本实例的关键。PHP 5.0 这种新型的字符串是以“<<<”开始，后面紧跟着字符串开始标记，之后为字符串的内容，最后以标记加分号（本实例以“mark”作为结束标记）结束。本函数一般用于对 HTML 代码的格式输出。

设计过程

创建 index.php 文件，通过 PHP5 新型字符串实现实例，其代码如下：

```
<?php
$str=<<<mark                                            //PHP5 新型字符串开始部分
<script language="javascript" type="text/javascript">   //JavaScript 代码
        alert("欢迎进入 PHP 编程世界！！！");            //输出提示
</script>                                               //结束标记
mark;                                                   //新型字符串介绍标记
echo $str;                                              //输出字符串
?>
```

秘笈心法

心法领悟 038：新型字符串。

笔者认为，新型字符串是将以往的在 HTML 标签中嵌套 PHP 标签的方式，做了一次有效的逆转。

实例 039　当数字遇到了字符串　高级

光盘位置：光盘\MR\02\039　趣味指数：★★★★

实例说明

在 PHP 手册中经常会看到一些返回值为 Boolean 的函数，函数说明的讲解可能是当符合条件时，返回值为 1，否则返回值为 0。按照正常的思维，返回值应该是 TRUE 和 FALSE，因为它们才是真正的 Boolean 类型。其实这些都是类型转换的应用。本实例介绍 PHP 中的数据类型的知识，展示数据是如何进行类型转换的，运行结果如图 2.7 所示。

自动类型转换：
10+我是字符串型数据=10
强制类型转换：
10+我是字符串型数据=10

图 2.7　当数字遇到了字符串

关键技术

虽然 PHP 是弱类型语言，但有时仍然需要用到类型转换。PHP 中的类型转换和 C 语言一样，非常简单。本实例的关键点是运用类型转换的特点进行输出。

（1）系统根据情况自动转换数据类型称为自动转换。

（2）用户在变量名称前将想要转换的数据类型写到括号内称为强制转换。

设计过程

（1）创建 PHP 脚本，通过 echo 语句输出想要转换的变量，其代码如下：

```
<?php
	$a = 10;                                    //整型数据
	$b = "我是字符串型数据";                    //字符串类型数据
	$e = $a + $b;
	$f = $b + $a;
	echo "自动类型转换：<br>";
	echo '10+我是字符串型数据='.$e."<br>";
	echo "强制类型转换：<br>";
	echo '10+我是字符串型数据='.(string)$e."<br>";
?>
```

（2）将文件存储于\MR\02\039\文件夹中，命名为 index.php。运行结果如图 2.7 所示。

秘笈心法

心法领悟 039：类型转换的相关说明。

（1）转换成整型。在将非整型数据转换成整型时，方法在变量前使用(integer)或者(int)，转换规则如下：

☑ 浮点型转换成整型。小数点后的数将被舍弃，如果浮点数超出整数取值范围，那么将无法得到有效的整型结果，结果可能是 0 或者整型的最小负数。

☑ 布尔型转换成整型。true 值将转换为 1，false 将转换为 0。

☑ 字符串型转换为整型。将会对字符串左侧的第 1 位进行判断。如果第 1 位是数字，则从第 1 位开始将读取到的数字转换成整型；如果第 1 位不是数字，则结果为 0。

（2）转换成浮点型。方法在变量前使用(float)，转换规则如下：

☑ 整型转换为浮点型。其结果不会发生变化。

☑ 布尔型转换为浮点型。同样 true 值将转换为 1，false 将转换为 0。

☑ 字符串型转换为浮点型。如果字符串中包含小数点“.”或科学记数法的“e”或者“E”中的任何一个字符，字符串被当作浮点型处理，否则，被视为整型。

（3）转换成字符串型。方法在变量前使用(string)，转换规则如下：

☑ 整型或者浮点型转换成字符串型。转换结果为其数值。

☑ 布尔型转换为字符串型。true 值将转换为字符串“1”，false 将转换为空字符串。

☑ 对象或者数组型转换成字符串型。那么转换结果为字符串对象或者字符串数组。

☑ 资源型转换成字符串型。转换结果为一个类似“Resource id #”的字符串。在“#”之后是 PHP 在运行时分配给该资源的标识代号。

（4）转换成布尔型。方法在变量前使用(boolean)或者(bool)。因为布尔型只包含两个值：true 和 false，所以其转换规则包括两个方面：

第一方面，转换结果为 false 的情况。

☑ 整型或者浮点型数 0。

☑ 空字符串和字符串“0”。

☑ 没有任何元素的空数组。

☑ 没有任何元素的对象。

☑ 特殊类型 NULL。

第二方面，转换结果为 true 的情况。排除在第一方面说明的情况，其他转换结果都为 true。

（5）转换成数组。方法在变量前使用(array)，转换规则是：将非数组型转换成与原变量数据类型相同的数组，数组中只有一个元素。

（6）转换成对象。方法在变量前使用(object)，转换规则是：将非对象型转换成一个新的对象，其中名为 scalar 的成员变量将包含原变量的值。

实例 040 PHP 程序员的基础——变量的应用

光盘位置：光盘\MR\02\040

高级

趣味指数：★★★★

实例说明

大楼盖得好不好主要在于地基是否牢固，变量就是 PHP 编程的地基。本实例介绍变量的定义与应用，通过 echo 语句实现打印“Hello World!!!”的小程序，运行结果如图 2.8 所示。

Hello World!!!

图 2.8　Hello World 程序

关键技术

本实例的关键点是用$符号定义变量，通过 echo()函数输出变量。PHP 中的变量使用“$”加变量名来表示，变量名是区分大小写的。使用变量时，只需对变量直接赋值即可。变量赋值，是指给变量一个具体的数据值，对于字符串和数字类型的变量，可以通过“=”来实现，其格式如下：

```
$name = value;
```

设计过程

（1）创建 PHP 脚本文件，定义字符串变量值为“Hello World!!!”，通过 echo 语句输出定义的变量，其代码如下：

```
<?php
	$str = "Hello World!!!";			//定义变量
	echo $str;				//输出变量
?>
```

（2）将该文件存储于\MR\02\040\文件夹下，命名为 index.php。运行结果如图 2.8 所示。

秘笈心法

心法领悟 040：变量的作用域。

变量有其自己的作用域，不同的作用域有不同的作用范围，就像清华大学学生的学生证不能在北京大学的图书馆借阅图书一样。

变量按其作用域可以分为全局变量、局部变量和静态变量，变量作用域的说明如表 2.4 所示。

表 2.4　变量作用域的说明

作　用　域	说　　明
全局变量	即被定义在所有函数以外的变量，其作用域是整个 PHP 文件，但是在用户自定义函数内部是不可用的。想在用户自定义函数内部使用全局变量，要使用 global 关键词声明，或者通过使用全局数组$globals 进行访问
局部变量	即在函数的内部定义的变量，这些变量只限于在函数内部使用，不能在函数外部使用
静态变量	能够在函数调用结束后保留变量值，当再次回到其作用域时，又可以继续使用原来的值。而一般变量在函数调用结束后，其存储的数据值将被清除，所占的内存空间被释放。使用静态变量时，先要用关键字 static 来声明变量（需要把关键字 static 放在要定义的变量之前）

注意：在函数的内部定义的变量，其作用域是所在函数。如果在函数外赋值，将被认为是完全不同的另一个变量。在退出声明变量的函数时，该变量及相应的值就会被撤销。

实例 041　打印系统环境变量信息 print_r($_ENV)

光盘位置：光盘\MR\02\041

高级　趣味指数：★★★★

实例说明

PHP 编程语言中预定义了很多有用的变量，这些变量往往起着非常重要的作用。本实例讲解预定义变量 $_ENV 的相关知识，打印输出系统的环境变量，运行结果如图 2.9 所示。

```
Array ( [ALLUSERSPROFILE] => C:\Documents and Settings\All Users
[CLASSPATH] => C:\Program Files\StormII\Codec;C:\Program Files\StormII
[CommonProgramFiles] => C:\Program Files\Common Files [COMPUTERNAME] =>
PC-201006101638 [ComSpec] => C:\WINDOWS\system32\cmd.exe
[FP_NO_HOST_CHECK] => NO [NUMBER_OF_PROCESSORS] => 1 [OS] => Windows_NT
[PATH] => C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem
[PATHEXT] => .COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH
[PROCESSOR_ARCHITECTURE] => x86 [PROCESSOR_IDENTIFIER] => x86 Family 15
Model 2 Stepping 9, GenuineIntel [PROCESSOR_LEVEL] => 15
[PROCESSOR_REVISION] => 0209 [ProgramFiles] => C:\Program Files
[SystemDrive] => C: [SystemRoot] => C:\WINDOWS [TEMP] => C:\WINDOWS\TEMP
[TMP] => C:\WINDOWS\TEMP [USERPROFILE] => C:\Documents and
Settings\LocalService [windir] => C:\WINDOWS [AP_PARENT_PID] => 292 )
```

图 2.9　预定义变量的应用

关键技术

PHP 提供的预定义变量有很多是非常实用的，通过这些预定义变量可以获取到用户会话、用户操作系统的环境和本地操作系统的环境等信息。其中，$_ENV 是执行环境提交至脚本的变量。

设计过程

（1）创建 PHP 脚本文件，通过 print_r()语句输出系统预定义变量，其代码如下：

```
<?php
	print_r($_ENV);		//打印系统的环境变量
?>
```

（2）将该文件存储于\MR\02\041\文件夹下，命名为 index.php，运行结果如图 2.9 所示。

秘笈心法

心法领悟 041：重要的系统预定义变量。

系统的预定义变量是比较常用的技术，读者必须牢记，尤其是$_FILES、$_POST、$_GET 和$_SESSION。

实例 042　使用可变变量输出"I Like PHP!"

光盘位置：光盘\MR\02\042

高级　趣味指数：★★★★

实例说明

变量包括预定义变量、可变变量等，预定义变量在实例 041 中已经做了简要介绍，本实例通过可变变量实现输出指定字符串，运行结果如图 2.10 所示。

```
I Like PHP!
```

图 2.10　使用可变变量输出

关键技术

可变变量是一种独特的变量，它允许动态改变一个变量名称，其工作原理是该变量的名称由另外一个变量的值来确定。实现过程是在变量前多加一个“$”符号。

设计过程

（1）创建 PHP 脚本文件。首先定义一个字符串变量，然后在第 1 个字符串变量的值前加一个“$”作为第 2 个变量的变量名，最后通过 echo 语句输出可变变量，其代码如下：

```
<?php
	$str_name = "str_name_1";			//定义变量
	$str_name_1 = "I Like PHP!";			//定义可变变量
	echo $$str_name;				//输出可变变量
?>
```

（2）将该文件存储于\MR\02\042\文件夹下，命名为 index.php。运行结果如图 2.10 所示。

秘笈心法

心法领悟 042：可变变量。

可变变量即是将两个变量的值使用一个变量命名进行操作，但是要注意被引用的变量名前要再加上一个“$”符号。

实例 043	使用转义字符输出特殊字符 光盘位置：光盘\MR\02\043	高级 趣味指数：★★★★

实例说明

在定义长字符串变量时，字符串本身往往包含一些特殊字符。例如字符串本身包含双引号（“”）。这时就需要使用转义字符对特殊字符进行转义。本实例通过转义字符“\”转义经特殊字符双引号（“”）包含的字符串，运行结果如图 2.11 所示。

"PHP是一门Web开发最火的编程语言"

图 2.11　使用转义字符输出特殊字符

关键技术

本实例的关键点是转义字符“\”的使用，它除了可以做转义字符外，还有一些其他功能，即可以将一些不可打印的字符显示出来，例如\a、\b、\e、\f、\n、\r 和\t 等。

设计过程

（1）创建 PHP 脚本文件，通过 echo 语句输出经转义字符转义的特殊字符，其代码如下：

```
<?php
	echo "\"PHP 是一门 Web 开发最火的编程语言\"";	//输出转义字符转义的字符串
?>
```

（2）将该文件存储于\MR\02\043\文件夹下，命名为 index.php。运行结果如图 2.11 所示。

秘笈心法

心法领悟 043：转义字符的功能。

转义字符的功能只有一个，就是将特殊字符转义成普通字符。

实例 044	使用常量指定 PI 的值计算圆的面积 光盘位置：光盘\MR\02\044	高级 趣味指数：★★★★

实例说明

常量是 PHP 编程基础的重要组成部分，其作用是定义一个不会改变的值。本实例通过计算圆的面积向用户说明常量是如何定义和使用的，运行结果如图 2.12 所示。

半径为10个单位的圆的面积314.15926

图 2.12　使用常量指定 PI 的值计算圆的面积

关键技术

PHP 中通常使用 define()函数来定义常量；使用 constant()函数动态获取常量值；使用 defined()函数判断一个常量是否已经定义；使用 get_defined_constants()函数获取所有当前已经定义的常量。其中，define()函数的语法如下所示：

define()：定义常量。

```
bool define ( string name, mixed value [, bool case_insensitive] );
```

参数说明：

string name：必选参数，常量名称，即标识符。

mixed value：必选参数，常量的值。

bool case_insensitive：可选参数，指定大小写是否敏感。设定为 true，表示不敏感。

设计过程

（1）创建 PHP 脚本文件，首先通过 define()函数定义常量，将数值“3.1415926”定义为常量名为 PI 的常量，然后定义数值型变量，将圆的半径设定为 10 个单位。最后通过 echo 语句输出圆面积，其代码如下：

```
<?php
    define("PI",3.1415926);                          //定义常量
    $r = 10;                                         //定义圆半径
    echo "半径为 10 个单位的圆的面积".PI*($r*$r);      //定义圆面积
?>
```

（2）将该文件存储于\MR\02\044\文件夹下，命名为 index.php。运行结果如图 2.12 所示。

秘笈心法

心法领悟 044：使用 define()定义常量的注意事项。

使用 define()定义的常量一旦定义就不能改变或者取消。

2.2　运　算　符

运算符是用来对变量、常量或数据进行计算的符号，是对一个值或一组值执行一个指定的操作。PHP 的运算符包括字符串运算符、算术运算符、赋值运算符、递增或递减运算符、位运算符、逻辑运算符、比较运算符和条件运算符。

实例 045 自定义数字的加密/解密算法

光盘位置：光盘\MR\02\045

高级

趣味指数：★★★★

实例说明

运算符是表达式的组成部分，没有运算符的表达式是不存在的。本实例通过使用算术运算符设置数字的加密/解密算法，运行结果如图 2.13 和图 2.14 所示。

数字口令： 确定

加密口令 1084.1415926

解密口令

图 2.13 口令加密

数字口令： 1081 确定

解密口令 1081

图 2.14 口令解密

关键技术

本实例的关键点是将数字进行算术运算进而得到加密/解密结果。算术运算（Arithmetic Operators）符号是处理四则运算的符号，在数字处理中应用得最多。

设计过程

（1）创建 PHP 脚本文件，首先定义常量，将数值“3.1415926”赋给常量名为 PI 的常量。然后编写自定义函数，在自定义函数方法体内定义加密或解密算法。最后，当表单提交后，调用对应的加密或解密算法，并输出结果，其代码如下：

```
<?php
	echo "<form action='' method='post'>";                          //输出表单
	echo "数字口令：<input name='text' type='text'>";
	echo "<input type='submit' name='sub' value='确定'>";
	echo "</form>";
	define("PI",3.1415926);                                        //定义常量
	function Encrypt($str){                                        //自定义加密算法
		return $str = $str + PI;                                   //将口令与常量相加
	}
	function Decrypt($str){                                        //自定义解密算法
		return $str = $str - PI;                                   //将加密后口令与常量相减
	}
	if($_POST[sub]){                                               //通过 post 方式传参
		echo "加密口令  ".Encrypt($_POST[text])."<br>";   //取得加密后的结果
		$_SESSION[pwd] = Encrypt($_POST[text]);                    //保存在 session 中
?>
	<a href='index.php?pwd=1'>解密口令</a>                          //超链接
<?php
	}
	if(isset($_GET[pwd])){                                         //判断地址栏中是否存在 pwd
		echo "解密口令  ".Decrypt($_SESSION[pwd]);        //口令解密
	}
?>
```

（2）将该文件存储于\MR\02\045\文件夹下，命名为 index.php。运行结果如图 2.13 和图 2.14 所示。

秘笈心法

心法领悟 045：运算符的优先级。

本实例并没有联合应用多种运算符，当联合应用时要注意运算符优先级的问题。所谓的优先级就是先进行哪种运算符的运算再进行哪种运算符的运算。为了方便用户，笔者将运算符的优先级进行归类，如表 2.5 所示。

表 2.5　运算符的优先级

优 先 级 别	运　算　符
1	Or，and，xor
2	=，+=，－=，*=，/=，.=，%=
3	||，&&
4	|，^
5	&，.
6	++，--（递增或递减运算符）
7	/，*，%
8	<<，>>
9	++，－－
10	+，－（正、负号运算符），!，~
11	==，!=，<>
12	<，<=，>，>=
13	?:
14	－>
15	=>

注意：在算术运算符中使用%求余时，如果被除数（$a）是负数，那么取得的结果也是一个负值。

实例 046　比较两个时间戳的大小　高级

光盘位置：光盘\MR\02\046　趣味指数：★★★★

实例说明

在一些程序中，经常需要将两个时间进行比较，但是由于时间是由年、月、日、时、分、秒组成的，比较起来很不方便，这时可以把时间转换成时间戳来进行比较。本实例通过 date()、strtotime()和 ceil()函数实现比较两个时间戳的大小，运行结果如图 2.15 所示。

```
1274924802 输出日期2010-05-27 09:46:42

1272988800 输出日期2010-05-05 00:00:00

距2010年5月5号已过去23天
```

图 2.15　比较两个时间戳的大小

关键技术

本实例主要是通过利用 strtotime()函数将时间转换为 UNIX 时间戳，再进行相关取整运算而实现的，其中涉及到的相关函数的语法如下：

（1）strtotime()函数。该函数预期接受一个包含英文日期格式的字符串，并尝试将其解析为 UNIX 时间戳：

```
int strtotime( string time [, int now]);
```

strtotime()函数有两个参数。如果参数 time 的格式是绝对时间，则 now 参数不起作用；如果参数 time 的格式是相对时间，其对应的时间由参数 now 来提供，若没有提供参数 now，则其对应的时间为当前时间。如果解

析失败，则返回 false。在 PHP5.1.0 之前的版本中，本函数在失败时返回-1。

（2）ceil()函数。该函数返回不小于 value 的下一个整数：

```
float ceil ( float value);
```

设计过程

（1）创建 PHP 脚本文件，首先对 PHP 语言中的时区进行设置，将时区更改为中国上海时区。然后，通过 strtotime()函数获取当前时间的 UNIX 时间戳，并与指定的时间戳进行算术运算。最后，将时间戳运算结果转换为天数，并通过 ceil()函数进行取整运算，然后输出结果，其代码如下：

```
<?php
    date_default_timezone_set("Asia/ShangHai");                //将格林威治时间设置为本地时间
    $a = strtotime("now");                                     //取得当前时间戳
    $b = strtotime("05 May 2010");                             //取得 2010 年 5 月 5 号的时间戳
    echo $a."\n";
    echo "输出日期".date("Y-m-d H:i:s",$a)."<br><br>";         //取得以$a 为时间点的时间
    echo $b."\n";
    echo "输出日期".date("Y-m-d H:i:s",$b)."<br><br>";         //取得以$b 为时间点的时间
    $c = ceil(($a - $b)/(3600*24));                            //计算相差天数
    echo "距 2010 年 5 月 5 号已过去".$c."天";
?>
```

（2）将该文件存储于\MR\02\046 文件夹下，命名为 index.php。运行结果如图 2.15 所示。

秘笈心法

心法领悟 046：时间戳的相关说明。

有效的时间戳的范围通常从 Fri, 13 Dec 1901 20:45:54 GMT 到 Tue, 19 Jan 2038 03:14:07 GMT（对应于 32 位有符号整数的最小值和最大值）。不是所有的平台都支持负的时间戳，因此日期范围就被限制为不能早于 UNIX 纪元。这意味着在 1970 年 1 月 1 日之前的日期将不能用在 Windows、一些 Linux 版本以及几个其他的操作系统中。

注意：在 UNIX 系统中，日期与时间表示为自 1970 年 1 月 1 日零点起到当前时刻的秒数，这种时间称为 UNIX 时间戳，用 32 位二进制表示。其中，1970 年 1 月 1 日零点称为 UNIX 世纪元。UNIX 时间戳提供了一种统一、简洁的时间表示方式，在不同的操作系统中均受到支持，同一时间在 UNIX 和 Windows 中均以相同的 UNIX 时间戳表示，所以不需要在不同的系统中进行转换。同时，UNIX 时间戳是一个时间差，与时区没有关系，无论当前 PHP 中使用的是何种时区，其 UNIX 时间戳是唯一的。

实例 047 使用条件运算符判断数字的奇偶性

光盘位置：光盘\MR\02\047

高级

趣味指数：★★★★

实例说明

条件运算符是用于执行或真或假运算的。本实例演示如何用三元运算判断数字的奇偶性，运行结果如图 2.16 所示。

0是偶数 1是奇数 2是偶数 3是奇数
5是奇数 6是偶数 7是奇数 8是偶数 9是奇数

图 2.16　用三元运算符判断奇数偶数

关键技术

本实例使用三元运算符判断数字的奇偶性，并且在判断数字奇偶性时使用了求余运算符。当期望数字与数字 2 做除法运算时，如果存在余数即不能被 2 整除，为奇数，否则为偶数。三元运算符的语法如下所示：

```
(expr) ? {statement1;} :  {statement2;}
```

三元运算符与 if…else…函数实现的功能完全相同。

设计过程

（1）创建 PHP 脚本文件，首先，通过 for 循环语句循环输出 0~9 的 10 个数字。然后，将得到的 10 个数字分别与 2 做求余运算并将运算结果作为三元运算的条件进行判断。最后，将判断的结果利用 echo()语句输出，其代码如下：

```
<?php
//做小于 10 的循环
    for($a = 0;$a < 10;$a++){
//用三元运算输出，如果没有余数则是偶数，否则为奇数
        echo   $a % 2 == 0 ?   $a."是偶数"."\n" : $a."是奇数"."\n";
}
?>
```

（2）将该文件存储于\MR\02\047\文件夹中，命名为 index.php。运行结果如图 2.16 所示。

秘笈心法

心法领悟 047：用 if…else…语句实现实例。

三元运算等同于 if…else…语句，下面就通过 if…else…语句实现本实例，其语法如下：

```
<?php
//做小于 10 的循环
for($a = 0;$a < 10;$a++){
//用 if…else…语句实现
        $b = 0;
        if($a% 2 ==0){
//设定条件，让$a 与 2 做求余运算
                echo "$a".'是偶数';
        }else{
                echo "$a".'是奇数';
        }
}
?>
```

实例 048　判断用户是否具有后台管理权限

光盘位置：光盘\MR\02\048

高级

趣味指数：★★★★

实例说明

逻辑运算符往往作为 if 等语句的条件出现。逻辑运算符有很多种而且功能各不相同。本实例介绍逻辑运算符的语法等相关知识，并通过逻辑运算符来演示用户是否具有后台管理权限，运行结果如图 2.17 和图 2.18 所示。



图 2.17　具有后台管理权限

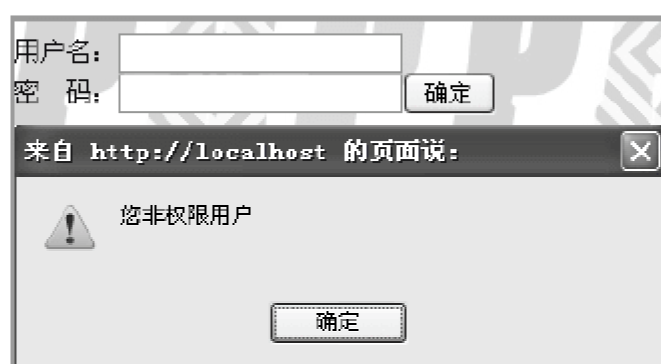


图 2.18　不具有后台管理权限

关键技术

本实例主要是通过逻辑运算符&&（逻辑与）将从表单中利用 POST 方法取得的由文本框输入的用户名和密码联系起来。用户名和密码必须同时为真，if 语句的结果才会为真，反之为假。

设计过程

（1）创建 PHP 脚本文件，首先，编写<form>表单，将表单中的 method 方法定义为 POST。然后，当单击按钮时，将 POST 方法接收的用户名和密码进行逻辑与运算。当 if 语句的结果为 True 时，输出“您具有管理员权限”的提示，否则输出“您非权限用户”的提示，其代码如下：

```
<?php
	echo "<form action="method='post'>";						//输出表单
	echo "用户名：<input type='text' name='text'><br>";
	echo "密  码：<input type='password' name='pwd'>";
	echo "<input type='submit' name='sub' value='确定'>";
	echo "</form>";
	if($_POST[sub]){									//通过 post 方式提交
		if($_POST[text] == "mr" && $_POST[pwd] == "mrsoft"){		//验证用户名密码
			echo "<script>alert('您具有管理员权限');</script>";		//符合条件输出提示
		}else{
			echo "<script>alert('您非权限用户');</script>";			//不符合条件输出提示
		}
	}
?>
```

（2）将该文件存储于\MR\02\048\文件夹中，命名为 index.php。运行结果如图 2.17 和图 2.18 所示。

秘笈心法

心法领悟 048：使用优先级的小技巧。

如果在涉及到运算符的优先级时不想总查找优先级表而本身又记忆不清的话，建议使用小括号把想要先运算的表达式括起来，这样问题就迎刃而解了。

实例 049 打印随机组合生日祝福语

光盘位置：光盘\MR\02\049

高级

趣味指数：★★★★

实例说明

在应用字符串时往往需要将两个字符串进行连接，这时候可以使用字符串运算符。本实例使用字符串运算符打印随机组合生日祝福语，运行结果如图 2.19 所示。

生日快乐祝你万事如意

图 2.19　打印随机组合生日祝福语

关键技术

字符串可以用“.”（点）字符串连接符连接，该连接符可以把两个或两个以上字符串连接成一个新的字符串。字符串的连接有两种形式，第 1 种是连接运算符“.”；第 2 种是连接赋值运算符“.=”。在实际的程序开发中，这两种连接方式都非常常用。

设计过程

创建 index.php 文件，首先，定义两个数组，将字符串信息保存在数组之中。然后，利用 rand 函数随机取

得数组中的两条信息并保存在变量中。最后，通过 echo 语句输出用连接运算符连接的变量，其代码如下：

```
<?php
        $arr = array("生日快乐","今天是你的出生日","同学们为你许愿");                     //定义数组
        $array = array("祝你万事如意","祝你生日快乐","祝你福如东海长流水寿比南山不老松");   //定义数组
        $rand = rand(0,2);                                                                  //定义随机数
        echo $arr[$rand].$array[$rand];                                                     //输出字符串
?>
```

秘笈心法

心法领悟 049：“+”号的相关说明。

使用过 C 或 JAVA 的读者应该注意，这里的“+”号，只做算术运算符使用，而不能做字符串运算符。

实例 050　打印 2000~2020 年的所有闰年

光盘位置：光盘\MR\02\050

高级

趣味指数：★★★★

实例说明

在 PHP 语言中，为了提升编程效率设置了一些自增自减运算符，这些运算符在循环语句中得到广泛应用。本实例简要介绍了自增自减运算符，并通过自增自减运算符修饰的变量来演示其应用，运行结果如图 2.20 所示。



图 2.20　打印 2000~2020 年的所有闰年

关键技术

递增或递减运算符有两种使用方法，一种是先将变量增加或减少 1 后再将值赋给原变量，称为前置递增或递减运算符（也称前置自增自减运算符）；另一种是将运算符放在变量后面，即先返回变量的当前值，然后变量的当前值增加或减少 1，称为后置递增或递减运算符（也称后置自增自减运算符）。本实例将递增递减运算符应用到 for 循环的变量之中，利用变量的变化完成期望次数的循环。

设计过程

创建 index.php 文件，首先，利用 for 循环定义循环变量，将变量的初始值设置为 2000，最大值小于 2020，并且要求变量做自增运算。然后，设置循环变量与 4 做求余运算。最后，把符合条件的数值通过 echo 语句输出，代码如下：

```
<?php
if($_POST[sub]){                                    //通过 post 接收参数
        for($a = 2000;$a <= 2020;$a++){             //通过循环定义循环变量
                if($a % 4 == 0){                    //将循环变量与 4 做求余运算
                        echo $a."  ";     //输出循环变量
                }
        }
}
?>
```

秘笈心法

心法领悟 050：自增自减运算符。

自增自减运算符在 PHP 循环结构中，特别是 for 循环结构中有着非常重要的地位。至于自增自减运算符前

置和后置的区别，请详见实例 051——前置运算符和后置运算符的区别。

实例 051 前置运算符和后置运算符的区别

光盘位置：光盘\MR\02\051

高级

趣味指数：★★★★

实例说明

前置运算符与后置运算符的应用方向与实例 050 相同，大多用在循环语句中。本实例应用自增自减运算符来修饰变量，通过显示结果来说明前置运算符与后置运算符的区别，运行结果如图 2.21 所示。

```
----------后置加运算---------
0 1 2 3 4 5 6 7 8 9 10
----------前置加运算---------
1 2 3 4 5 6 7 8 9 10 11
----------前置减运算---------
9 8 7 6 5 4 3 2 1 0 -1
----------后置减运算---------
10 9 8 7 6 5 4 3 2 1 0
```

图 2.21　前置运算符与后置运算符的区别

关键技术

本实例的关键点是自增自减运算符的灵活运用，相关说明详见实例 050。在使用前置减运算符时，要注意给定的初始值，否则会出现本例的问题，即前置减运算输出负数。

设计过程

（1）创建 PHP 脚本文件，首先，通过 for 循环语句定义循环变量，然后，通过 echo 语句输出循环变量，其代码如下：

```
<?php
    $a = 0;                                          //自定义变量并给定初始值
    echo "----------后置加运算---------<br>";
    for($b = 0;$b <= 10;$b++){                       //使用 for 循环语句
        echo   $a++." ";                             //使用后置运算符
    }
    echo "<br>----------前置加运算---------<br>";
    $d = 0;
    for($b = 0;$b <= 10;$b++){
        echo ++$d." ";                               //使用后置运算符
    }
    echo "<br>----------前置减运算---------<br>";
    $f =10;
    for($b = 10;$b >= 0;$b--){
        echo --$f." ";                               //使用前置运算符
    }
    echo "<br>----------后置减运算---------<br>";
    $g =10;
    for($b = 10;$b >= 0;$b--){
        echo $g--." ";                               //使用后置运算符
    }
?>
```

（2）将该文件存储于\MR\02\051\文件夹下，命名为 index.php。运行结果如图 2.21 所示。

秘笈心法

心法领悟 051：前置运算符。

前置运算符是在程序第一次运行时就进行自增或自减运算，而后置运算符是在程序运行完一次之后再进行自增或自减运算。

实例 052　使用位运算对数字进行加密和解密　高级

光盘位置：光盘\MR\02\052　趣味指数：★★★★

实例说明

位运算符也是 PHP 运算符中不可或缺的一种。本实例通过位运算符实现对数字进行加密，运行结果如图 2.22 和图 2.23 所示。

数字口令：111 确定
加密口令 888
解密口令

图 2.22　使用位运算符对数字进行加密

数字口令：111 确定
解密口令 111

图 2.23　使用位运算符对数字进行解密

关键技术

位运算符是指对二进制位从低到高位对齐后进行运算，PHP 中的位运算符如表 2.6 所示。本实例中，利用自定义函数，将文本框信息与定义的常量做指定的位运算，并返回计算结果，从而将文本框信息进行加密或解密处理。

表 2.6　位运算符

符　　号	作　　用	举　　例
&	按位与	$m&$n
|	按位或	$m\|$n
^	按位异或	$m^$n
~	按位取反	$m~$n
<<	向左移位	$m<<$n
>>	向右移位	$m>>$n

设计过程

（1）创建 PHP 脚本文件，首先，编写<form>表单并定义常量，将数值“3.1415926”赋给常量 PI。然后，定义自定义函数，在函数体内部，将文本框信息与常量 PI 做移位运算并返回结果。最后，通过提交数据和单击按钮实现输出加密、解密算法的结果，其代码如下：

```
<?php                                                          //定义表单
    echo "<form action='' method='post'>";
    echo "数字口令：<input name='text' type='text'>";
    echo "<input type='submit' name='sub' value='确定'>";
    echo "</form>";
    define("PI",3.1415926);                                    //定义常量
    function Encrypt($str){                                    //自定义加密算法
        return $str = $str << PI;
    }
    function Decrypt($str){                                    //自定义解密算法
        return $str = $str >> PI;
    }
    if($_POST[sub]){                                           //通过 post 方式传参
        echo "加密口令  ".Encrypt($_POST[text])."<br>";      //输出口令
```

```
                $_SESSION[pwd] = Encrypt($_POST[text]);                         //将口令保存在 session 中
?>
                <a href='index.php?pwd=1'>解密口令</a>                          //超链接
<?php
        }
        if(isset($_GET[pwd])){                                                  //判断地址栏是否存在此参数
                echo "解密口令  ".Decrypt($_SESSION[pwd]);             //输出解密结果
        }
?>
```

（2）将该文件存储于\MR\02\052\文件夹下，命名为 index.php。运行结果如图 2.22 和图 2.23 所示。

秘笈心法

心法领悟 052：位运算符的相关说明。

位运算符允许对整型数据中指定的位进行置位。如果左右参数都是字符串，则位运算符将操作字符的 ASCII 值。

2.3 包 含 语 句

引用外部文件可以减少代码的重用性，是 PHP 编程的重要技巧。PHP 提供了 4 个非常简单有用的语句，它们允许重新使用任何类型的代码。使用任意一个语句均可以将一个文件载入 PHP 脚本中，为网站创建统一风格的页面，从而更加轻松地编写和维护代码。

实例 053	提高代码重用率 光盘位置：光盘\MR\02\053	高级 趣味指数：★★★★

实例说明

当需要在文件外部引入新文件时就要借助 include()或 require()等包含语句。本实例通过 require()包含语句实现代码重用，运行结果如图 2.24 所示。

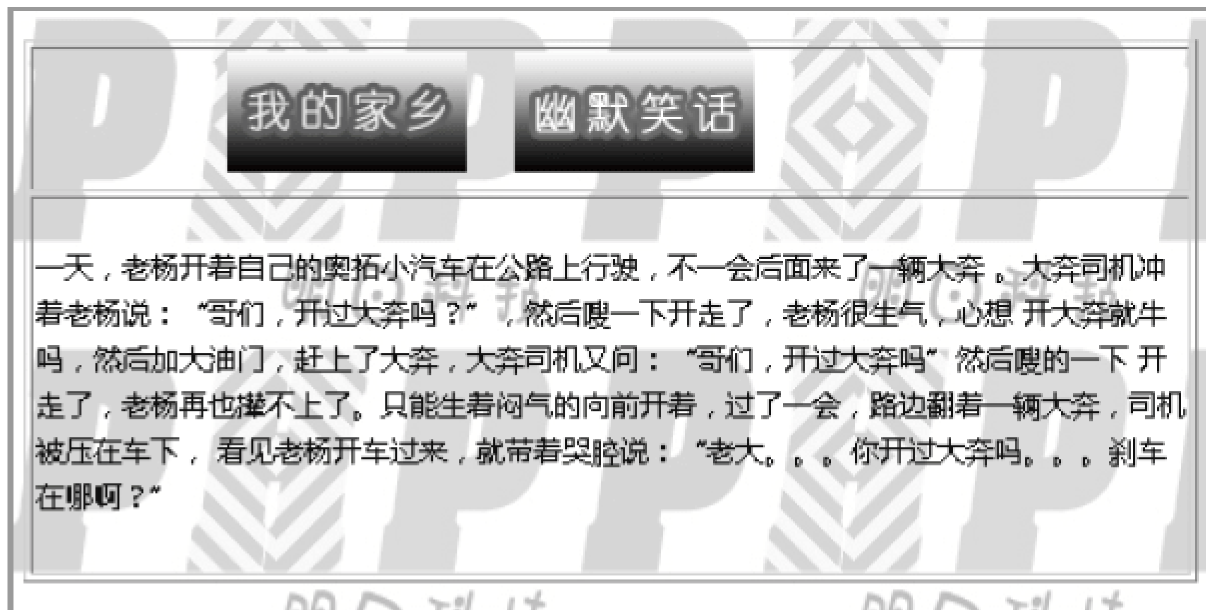


图 2.24　提高代码重用率

关键技术

本实例通过 require()语句包含并运行指定文件。PHP 文件被执行之前，PHP 解析器会用被引用的文件的全部内容替换 require()语句，然后与 require()语句之外的其他语句组成新的 PHP 文件。最后再按新的 PHP 文件执行程序代码，其语法如下：

```
void require(string filename);
```

参数说明：

filename：是指定的完整路径文件名。

设计过程

（1）创建 inc.php 文件，建立两个字符串表达式，代码如下：

```
<?php
$str1 ="大连是一个美丽的地方，它三面环海，                              //声明两个字符串变量
背依大山，就像一个乘风破浪的水手，
就算再大的风浪，也依然昂首挺胸不屈不挠";
$str2 = "一天，老杨开着自己的奥拓小汽车在公路上行驶，不一会后面来了一辆大奔
。大奔司机冲着老杨说：“哥们，开过大奔吗？”，然后嗖一下开走了，老杨很生气，心想
开大奔就牛吗，然后加大油门，赶上了大奔，大奔司机又问：“哥们，开过大奔吗”然后嗖地一下
开走了，老杨再也撵不上了。只能生着闷气地向前开着，过了一会，路边翻着一辆大奔，司机被压在车下，
看见老杨开车过来，就带着哭腔说：“老大。。。你开过大奔吗。。。刹车在哪啊？”";
?>
```

（2）创建 index.php 文件，通过 require()语句包含文件 inc.php，代码如下：

```
<?php
        require("inc.php");
        switch($_GET['link']){                              //接收地址栏的值
                case "我的家乡";                            //如果值等于我的家乡
                echo $str1;                                 //输出字符串 1
                break;                                      //停止执行语句
                case "幽默笑话";                            //如果值等于幽默笑话
                echo $str2;                                 //输出字符串 2
                break;                                      //停止执行语句
        }
?>
```

秘笈心法

心法领悟 053：使用 require()语句的注意事项。

因为使用 require()语句相当于将另一个源文件的内容完全复制到本文件中，所以一般将该语句放在源文件的起始位置，用于引用需要使用的公共函数和公共类文件等。

实例 054　包含数据库连接文件

光盘位置：光盘\MR\02\054

高级

趣味指数：★★★★

实例说明

一个程序可能要与数据库多次交互，所以数据库连接信息要单独保存在某一文件中，这也是代码重用的一种体现。本实例通过 include ()语句包含数据库文件，运行结果如图 2.25 所示。

编号：1　姓名：杨明
编号：2　姓名：潘凯华
编号：3　姓名：杨明
编号：4　姓名：潘凯华
编号：5　姓名：刘中华
编号：6　姓名：刘中华

图 2.25　包含数据库连接文件

关键技术

include()语句与 require()语句在作用上是完全相同的，但是使用 include()语句引用外部文件时，只有代码执行到该语句时才将外部文件引用进来并读取文件的内容，当所引用的外部文件发生错误时，系统只给出一个警

告，整个 PHP 文件则继续向下执行。include()语句的语法如下：

```
void include(string filename);
```

参数说明：

filename：是指定的完整路径文件名。

设计过程

（1）创建 index.php 文件，通过 include()语句包含数据库文件，并执行查询操作，输出结果，其代码如下：

```
<?php
include ("inc.php");                                    //包含数据库连接文件
$sql = "select * from tb_include";                      //sql 语句
$rs = mysql_query($sql);                                //执行查询操作
while($rst = mysql_fetch_array($rs)){                   //将查询结果循环输出
        echo '编号：'.$rst[0].'  姓名：'.$rst[1]."<br>";
}
?>
```

（2）创建数据库连接文件 inc.php，其代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111");        //连接 MYSQL
mysql_select_db("db_database02",$conn);                 //连接数据库
mysql_query("SET NAMES GBK");                           //设置编码格式
?>
```

秘笈心法

心法领悟 054：使用 include()语句的注意事项。

include ()语句当包含一个不存在的文件时会产生警告。

实例 055	包含网站头文件 光盘位置：光盘\MR\02\055	高级 趣味指数：★★★★

实例说明

在设计网站的过程中往往都是先用 table 或 div 标签将整个网站进行布局，然后通过 include_once()等语句将各部分内容包含进来。本实例介绍使用 include_once()语句包含网站头文件，运行结果如图 2.26 所示。

图 2.26　包含网站头文件

关键技术

PHP 程序开发中，在使用 include_once()语句时，应该明确其与 include()语句的区别。include_once()语句会在导入文件前检测该文件是否在该页面的其他部分被引用过，如果有，则不会重复引用该文件，即程序只引用一次。例如，要导入的文件中存在一些自定义函数，如果在同一个程序中重复导入这个文件，在第 2 次导入时便会发生错误，因为 PHP 不允许相同名称的函数被重复声明。

设计过程

（1）创建 index.php 文件，通过 include_once()语句包含文件 inc.php，其代码如下：

```
<table><tr><td><?php include_once("inc.php");?></td></tr></table>                    //在 table 标签中嵌入 php 包含语句
```

（2）创建 inc.php 文件，将图片作为<body>标签的背景图，其代码如下：

```
<title>包含头文件</title>
<body background="../image/1.jpg" style="font-size:14px; color:#000099; font-family:'微软雅黑'">    //背景
<img src="image/1.jpg">                                                                          //引入图片
</body>
```

（3）运行本实例，运行结果如图 2.26 所示。

秘笈心法

心法领悟 055：使用 include_once()语句的注意事项。

include_once()语句应用于在脚本执行期间同一个文件有可能被包含超过一次的情况下，确保指定文件只被包含一次，从而避免函数重复定义，变量重新赋值。

实例 056　包含网站尾文件

光盘位置：光盘\MR\02\056

高级

趣味指数：★★★★

实例说明

require()语句和 include()语句在作用上没有本质区别，用户可根据个人习惯选择使用。本实例通过使用 require()语句包含网站尾文件，运行结果如图 2.27 所示。



图 2.27　包含网站尾文件

关键技术

本实例的关键点是 rerquire()语句的运用。require()语句的详细说明请参见实例 053。

设计过程

（1）创建 index.php 文件，编写 PHP 标记，通过 require()语句包含文件 inc.php，其代码如下：

```
<?php
    require('inc.php');                                                                  //包含尾文件
?>
```

（2）创建 inc.php 文件，定义网站尾文件，利用<p>标签将明日科技公司的相关信息包含进来，其代码如下：

```
<table width="800" height="129" border="1" bordercolor="#CCCCCC" bordercolorlight="#CCCCCC" cellspacing="0">
  <tr>
    <td width="800"><div align="center">                                                  //定义表格
      <p>长春市明日科技有限公司</p>
      <p>电话：0431-84978981</p>
      <p>传真：0431-84978981</p>
      <p>email:www.mrsoft.com</p>
    </div></td>
```

```
    </tr>
</table>
```

（3）运行本实例，运行结果如图 2.27 所示。

秘笈心法

心法领悟 056：使用 require()语句的注意事项。

require()语句包含不存在的文件时，会出现一个致命错误。

实例 057	包含网站的主文件 光盘位置：光盘\MR\02\057	高级 趣味指数：★★★★

实例说明

本实例通过使用 require_once()语句包含网站主文件，运行结果如图 2.28 所示。



图 2.28　包含网站主文件

关键技术

require_once()语句是 require()语句的延伸，其功能与 require()语句基本类似，不同的是在应用 require_once()语句时会先检查要引用的文件是不是已经在该程序中的其他地方被引用过，如果有，则不再重复调用该文件，例如，同时应用 require_once()语句在同一个页面中引用了两个相同文件，那么在输出时只有第 1 个文件被执行，第 2 次引用的文件不会被执行。

设计过程

（1）创建 index.php 文件，通过 switch()语句接收地址栏传递的参数，根据参数传递的数据引入通过 require_once()语句包含的文件，其代码如下：

```
<?php
    switch($_GET['link']){                    //通过 get 方式接收地址栏参数
        case "主页";
        require_once('inc.php');              //包含文件 inc.php
        break;
        case "注册";
```

```
            require_once('inc_1.php');          //包含文件 inc_1.php
            break;
            case "浏览";
            require_once('inc_2.php');          //包含文件 inc_2.php
            break;
            default:
            include('inc.php');
        }
?>
```

（2）运行本实例，运行结果如图 2.28 所示。

秘笈心法

心法领悟 057：使用 require_once()语句的注意事项。

require_once()语句与 require()语句一样，当包含不存在的文件时，会出现一个致命错误。

2.4　条 件 语 句

条件控制语句就是以一定的条件作为依据，根据判断的结果确定执行哪一部分代码，而不执行与该部分代码并列的其他代码。

在 PHP 中，条件控制语句可分为以下两种类型：

☑　if 条件控制语句。

☑　switch...case 分支控制语句。

实例 058	员工生日提醒 光盘位置：光盘\MR\02\058	高级 趣味指数：★★★★

实例说明

在 PHP 应用中，类似生日提醒等功能的程序随处可见，主要是因为这样的程序有一定的定时效果。本实例通过 if()语句和 foreach()循环语句实现一个员工生日提醒的小程序，如图 2.29 所示。

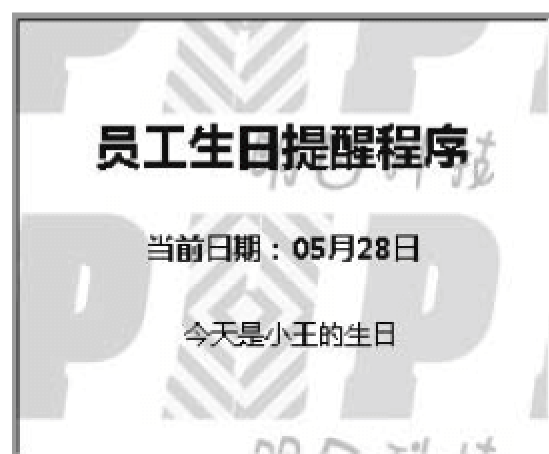


图 2.29　员工生日提醒

关键技术

本实例的关键点是利用 if 选择语句，将文本框中的日期与数组中员工生日日期进行比较。

（1）foreach()语句

foreach()语句仅能用于数组，当试图将其用于其他数据类型或者一个未初始化的变量时会产生错误。

```
foreach (array_expression as $value) statement;
```

参数 array 表示要遍历的数组，expression 表示键值，$value 表示键值的对应值，statement 表示语句块。

（2）if()语句

if()语句对某段程序的执行附加一个条件，如果条件成立，就执行这段程序；否则就跳过这段程序。

```
if(expr) statement;
```

参数 expr 表示判定条件，statement 表示语句块。

（3）array()函数

array()函数返回根据参数建立的数组。参数可以用 => 运算符给出索引。

```
array array ( [mixed ...]);
```

设计过程

（1）创建 index.php 文件。首先，根据数组的特点将键值与对应值存储在数组之中，然后，利用 foreach 语句遍历数组，将取得的 value 值与当前日期进行比较。最后，将符合条件的值进行输出，其代码如下：

```
<?php
    $a = array("小杨"=>"05 月 30 日","小王"=>"05 月 28 日","小张"=>"05 月 29 日");        //定义数组
        foreach($a as $key => $value){                                                //用 foreach 语句循环数组
            if(date("m 月 d 日") == $value){                                          //用 if 条件语句进行比较
                echo "  今天是".$key."的生日";                               //输出结果
            }
        }
?>
```

（2）运行本实例，运行结果如图 2.29 所示。

秘笈心法

心法领悟 058：使用 foreach()语句的相关说明。

foreach()语句所操作的数据是指定数组的一个副本，而不是该数组本身。因此数组指针不会被 for each()结构改变，对返回的数组单元的修改也不会影响源数组。不过源数组的内部指针的确在处理数组的过程中向前移动了。假定 foreach 循环运行到结束，源数组的内部指针将指向数组的结尾。foreach()语句不支持用“@”来禁止错误信息。

实例 059	考试成绩评定标准 光盘位置：光盘\MR\02\059	高级 趣味指数：★★★★

实例说明

本实例通过 if()条件语句对文本框中输入的考试成绩进行评定和选择，运行结果如图 2.30 所示。



图 2.30 考试成绩评定标准

关键技术

if()语句是最简单的条件判定语句，它对某段程序的执行附加一个条件，如果条件成立，就执行这段程序；否则就跳过这段程序去执行下面的程序。

本实例的关键点是 if()语句的灵活运用。当 expr 的值为 TRUE 时，执行 statement1 语句。

```
if(expr){statement1;}
```

参数说明：

expr：条件判断语句。

statement：符合条件的执行代码。

设计过程

创建 index.php 文件，当单击“评定”按钮时，程序自动利用 POST 方法接收数据，并且利用 if()语句对接收的数据进行判断，再用 echo 语句输出结果，其代码如下：

```
<?php
	if($_POST[sub]){                                        //通过 post 方式传递参数
		if($_POST[text] == 100){                              //当成绩为 100 时
			echo "成绩优秀";                                  //输出
		}
		if($_POST[text] >= 60 and $_POST[text] < 100){       //当成绩大于等于 60 小于 100 时
			echo "成绩良好";                                  //输出
		}
		if($_POST[text] < 60){                               //当成绩小于 60 时
			echo "不及格";                                    //输出
		}
	}
?>
```

秘笈心法

心法领悟 059：if()语句的变体。

if()语句有多种变体，因此其应用范围非常广泛，在后面的很多实例中都可以找到它的身影。

实例 060	控制登录用户权限 光盘位置：光盘\MR\02\060	高级 趣味指数：★★★★

实例说明

用户在浏览一些网站时，网站本身会将用户定义为 VIP 会员或者游客，会员可以享有更多的权限，而游客对网站的浏览有权限限制，下面就来实现这一功能。本实例通过 if…else…语句实现对登录用户权限的控制，运行结果如图 2.31 所示。

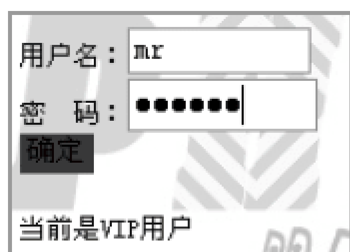


图 2.31　控制登录用户权限

关键技术

大多时候，总是需要在满足某个条件时执行一条语句，不满足时执行另一条语句。为了适应这样的一个需求，用户可以使用 if()语句的 else 子句，其语法如下：

```
if(expr){statement1;}else{statement2;}
```

如果表达式 expr 的值为 TRUE，则执行 statement1 语句；如果表达式 expr 的值为 FALSE，则执行 statement2 语句。

设计过程

（1）创建 PHP 脚本文件，当单击“确定”按钮时，程序通过 POST 方法接收传递过来的数据，并且通过 if…else…选择语句设定条件，当符合条件时利用 echo 语句输出“当前是 VIP 用户”，否则输出“您只是一个游

客”，其代码如下：

```
<?php
    if($_POST['sub']){
        //判断通过 POST 接收的参数结果是否符合条件
        if($_POST['user'] == 'mr' and $_POST['pwd'] == 'mrsoft'){
            echo "当前是 VIP 用户";                          //符合条件的结果
        }else{
            echo "您只是一个游客";                           //不符合条件的结果
        }
    }
?>
```

（2）将该文件存储于\MR\02\060\文件夹中，命名为 index.php。运行结果如图 2.31 所示。

秘笈心法

心法领悟 060：使用 if…else…语句的优点。

if…else…语句是 if 语句的另一种形式，结构上相对 if 语句要清晰。

实例 061	网页框架的制作 光盘位置：光盘\MR\02\061	高级 趣味指数：★★★★

实例说明

类似于留言板等一些功能较单一的网站，可以通过 switch 语句制作网页框架将所有内容包含到主页当中。本实例通过 switch()语句、include()语句来演示网页框架的制作，运行结果如图 2.32 所示。

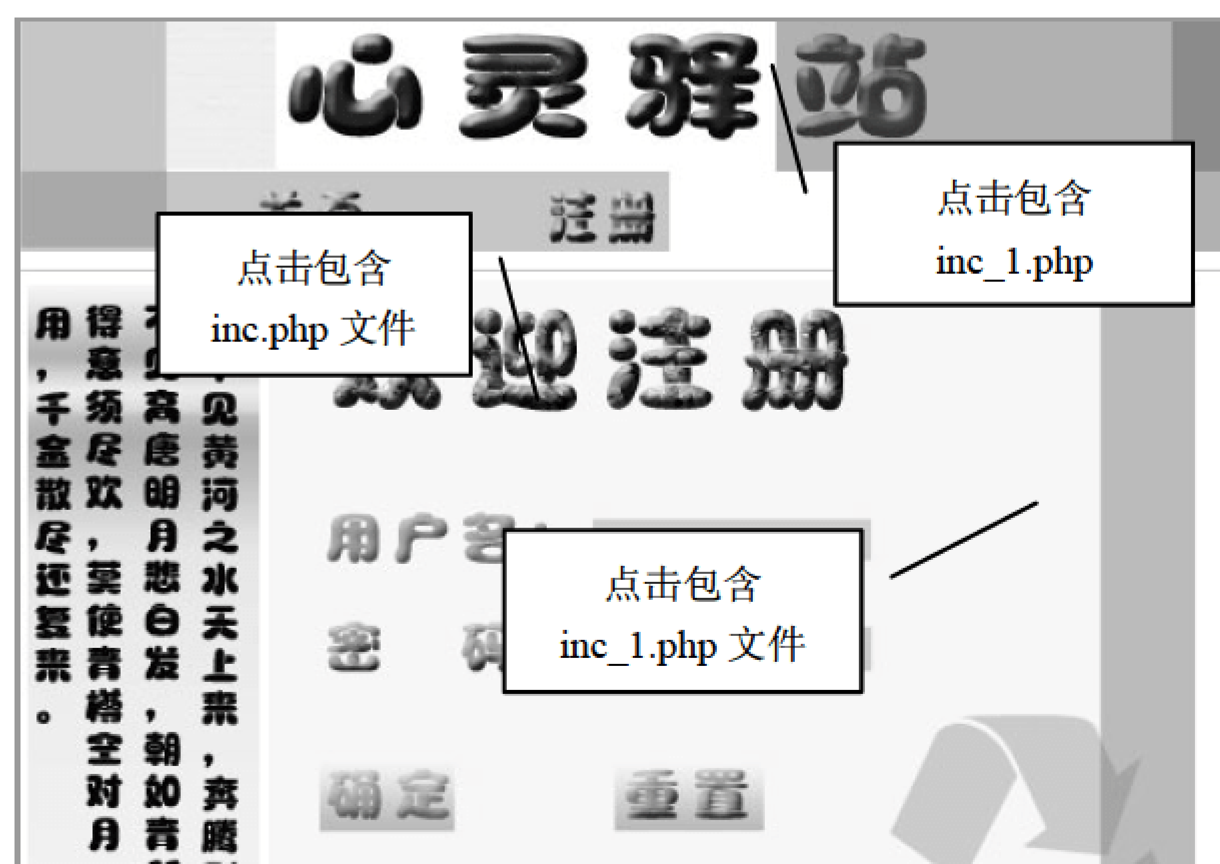


图 2.32　网页框架的制作

关键技术

switch()语句和 break 语句的灵活运用是本实例的关键。

（1）switch()语句和具有同样表达式的一系列的 if 语句相似。很多场合下需要把同一个变量（或表达式）与很多不同的值比较，并根据它等于哪个值来执行不同的代码，其语句的格式如下：

```
switch(variable){
case value1:
        statement1;
        break;
case value2:
```

```
...
default:
        default statement n;
}
```

switch()语句根据 variable 的值，依次与 case 中的 value 值相比较，如果不相等，继续查找下一个 case；如果相等，就执行对应的语句，直到 switch 语句结束或遇到 break 为止。一般 switch()语句最终都有一个默认值 default，如果在前面的 case 中没有找到相符的条件，则输出默认语句（与 else 语句类似）。

（2）break 语句用于结束当前 for、foreach、while、do...while 或者 switch 结构的执行。

设计过程

（1）创建 PHP 脚本文件，首先，利用 switch()语句获取从地址栏传递的数据，并且将数据与 switch()语句中的 case 做比较，当地址栏传递的数据与 case 中的数据相等时，包含指定文件，并且利用 break 语句跳出 switch()语句，其代码如下：

```
<?php
        switch($_GET['link']){                          //通过地址栏接收参数
                case "主页";                             //如果值为主页
                include('inc.php');                     //包含“inc.php”文件
                break;                                  //结束执行 switch 语句
                case "注册";                             //如果值为注册
                include('inc_1.php');
                break;
                case "浏览";                             //如果值为浏览
                include('inc_2.php');
                break;
                default:                                //默认情况下，包含文件“inc.php”
                include('inc.php');
        }
?>
```

（2）将该文件存储于\MR\02\061\文件夹下，命名为 index.php。运行结果如图 2.32 所示。

秘笈心法

心法领悟 061：switch()语句的实现原理。

switch()语句在执行时，即使遇到符合要求的 case 语句段，也会继续往下执行，直到 switch 结束。为了避免这种浪费时间和资源的行为，一定要在每个 case 语句段后添加 break 跳转语句跳出当前循环。

实例 062	图片验证码 光盘位置：光盘\MR\02\062	高级 趣味指数：★★★★

实例说明

验证码的作用是防止用户恶意注册，而降低数据库和网站本身的性能。本实例应用 switch()语句通过 PHP 图像函数来演示图片验证码的制作，运行结果如图 2.33 所示。



图 2.33　图片验证码

关键技术

本实例的关键点是验证码函数和 switch()语句的联合应用，其应用到的函数如下：

（1）header()函数

当参数为 Content-type:image/png 时使用 header()函数表示将图片输出到浏览器。

```
void header ( string string [, bool replace [, int http_response_code]]);
```

（2）imagecreatetruecolor()函数

新建一个真彩色图像，imagecreatetruecolor()函数返回一个图像标识符，代表了一幅大小为 x_size 和 y_size 的黑色图像。

```
resource imagecreatetruecolor ( int x_size, int y_size);
```

（3）imagecolorallocate()函数

为一幅图像分配颜色，imagecolorallocate()函数返回一个标识符，代表了由给定的 RGB 成分组成的颜色。image 参数是 imagecreate() 函数的返回值；red、green 和 blue 分别是所需要的颜色的红、绿、蓝成分。这些参数是 0~255 的整数或者十六进制的 0x00~0xFF。imagecolorallocate() 必须被调用以创建每一种用在 image 所代表的图像中的颜色。

```
int imagecolorallocate ( resource image, int red, int green, int blue);
```

（4）imagettftext()函数

用 TrueType 字体向图像写入文本，imagettftext()函数是将字符串 text 添加到 image 所代表的图像上，从坐标（x，y）（左上角为（0, 0））开始，角度为 angle，颜色为 color，使用 fontfile 所指定的 TrueType 字体文件。根据 PHP 所使用的 GD 库的不同，如果 fontfile 没有以 '/'开头，则 '.ttf' 将被加到文件名之后并且会搜索库定义字体路径。

```
array imagettftext ( resource image, int size, int angle, int x, int y, int color, string fontfile, string text);
```

（5）imagegif()函数

以 GIF 格式将图像输出到浏览器或文件，imagegif()函数从 image 图像以 filename 为文件名创建一个 GIF 图像。image 参数是 imagecreate() 函数的返回值。

```
int imagegif ( resource image [, string filename]);
```

（6）int imagestring()函数

水平地画一行字符串，int imagestring()函数用 col 颜色将字符串 s 画到 image 所代表的图像的（x，y）坐标处（图像的左上角为（0, 0））。如果 font 是 1、2、3、4 或 5，则使用内置字体。

```
int imagestring ( resource image, int font, int x, int y, string s, int col);
```

设计过程

（1）创建 index.php 文件，通过 GET 方式接收由地址栏传递的数据并且将参数与 case 的值进行比较，当两者相等时引入指定图片，通过 switch()语句进行操作，其代码如下：

```
<?php
	switch($_GET['link']){					//取得通过地址栏传递的参数
		case "文字";					//当参数等于文字时
?>
		<img src="yzm_1.php">				//引入图片
<?php
		break;						//结束执行 switch()语句
		case "数字";					//当参数等于数字时
?>
		<img src="yzm_2.php">				//引入图片
<?php
		break;						//结束执行 switch()语句
		default:					//默认情况下
?>
		<img src="yzm_2.php">
<?php
}
?>
```

（2）创建 yzm_1.php 文件，应用图像处理函数生成文字验证码，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
header("Content-type:image/gif");                                        //设置页面类型
$image = imagecreatetruecolor(80,30);                                    //创建画布
$font = 'Font/FZHCJW.TTF';                                               //定义字体
$bg = imagecolorallocate($image,255,255,255);                            //定义背景颜色
$color = imagecolorallocate($image,255,0,255);
imagettftext($image,20,0,8,20,$color,$font,"mrsoft");                    //输出文字
imagegif($image);                                                        //生成图像
?>
```

（3）创建 yzm_2.php 文件，应用图像处理函数生成数字验证码，其代码如下：

```
<?php
header("Content-type:image/gif");                                        //设置页面格式
$image = imagecreatetruecolor(40,20);                                    //生成画布
$bg = imagecolorallocate($image,225,225,0);                              //定义背景颜色
$color = imagecolorallocate($image,255,0,255);                           //定义图像颜色
for($b = 0;$b <4; $b ++){                                                //生成随机验证码
        $a .= rand(1,9);
}
imagestring($image,3,8,5,$a,$color);                                     //输出验证码
imagegif($image);                                                        //生成图像
?>
```

秘笈心法

心法领悟 062：开启 GD2 支持。

PHP 5.0 中 GD2 函数库已经作为扩展被默认安装，但目前有些版本中，还需要对 php.ini 文件进行设置来激活 GD2 函数库。用文本编辑工具，如记事本等程序打开 php.ini 文件，将文件中的“;extension=php_gd2.dll”选项中的分号“;”删除，保存修改后的文件，并重新启动 Apache 服务器即可激活 GD2 函数库。

实例 063	健康生活提醒 光盘位置：光盘\MR\02\063	高级 趣味指数：★★★★

实例说明

一些手机在开机时会出现开机问候语或者今日日程提示。本实例就是设计类似这样的程序，通过 switch()语句根据当前日期，给出健康生活提示信息，运行结果如图 2.34 所示。



图 2.34　健康生活提醒

关键技术

本实例的关键点是利用 switch()语句，将地址栏传递的数据与 switch 中 case 值进行比较，符合条件的利用 echo 语句输出对应数据。

设计过程

（1）创建 PHP 脚本文件。首先，通过 date()函数取得当前是星期几，然后，利用 switch()语句进行操作，并通过 echo 语句输出信息提示，其代码如下：

```
<?php
        $a = date("l");                                                  //取得当前是星期几
```

```
    switch($a){
        case "Monday";                                  //当前如果是星期一
        echo "今天是星期一，一周忙碌生活开始了";         //输出
        break;
        case "Tuesday";                                 //当前如果是星期二
        echo "今天是星期二，电视台下午两点以后部分休息";  //输出
        break;
        case "Wednesday";                               //当前如果是星期三
        echo "今天是星期三，下午有乒乓球比赛";           //输出
        break;
        case "Thursday";                                //当前如果是星期四
        echo "今天是星期四，晚上有 NBA 的重播";          //输出
        break;
        case "Friday";                                  //当前如果是星期五
        echo "今天是黑色星期五。。。。。。";             //输出
        break;
        case "Saturday";                                //当前如果是星期六
        echo "今天是星期六，明天就放假了哈哈哈";         //输出
        break;
        case "Sunday";                                  //当前如果是星期日
        echo "今天是星期天，可以玩上一整天";             //输出
    }
?>
```

（2）将该文件存储于\MR\02\063\文件夹中，命名为 index.php。运行结果如图 2.34 所示。

秘笈心法

心法领悟 063：比较 switch()语句与 if…else…语句。

switch()语句与 if…else…语句在功能上基本相同，但是 switch()语句更加灵活，格式更加鲜明，结构更加清晰。

2.5 循环控制

在实际应用中，经常会遇到一些操作并不复杂，但需要反复处理的问题，为此，PHP 提供了循环语句来实现循环结构的程序设计。循环控制语句是指能够按照一定的条件重复执行某段功能代码的代码结构。循环控制语句分为以下 4 种：

☑ while 循环语句。
☑ do...while 循环语句。
☑ for 循环语句。
☑ foreach 循环语句。



实例 064	员工生日列表 光盘位置：光盘\MR\02\064	高级 趣味指数：★★★★

实例说明

while 循环结构的作用是重复执行一段代码或完成相同的动作。本实例通过 while()语句循环输出数组中存储的员工生日信息，运行结果如图 2.35 所示。

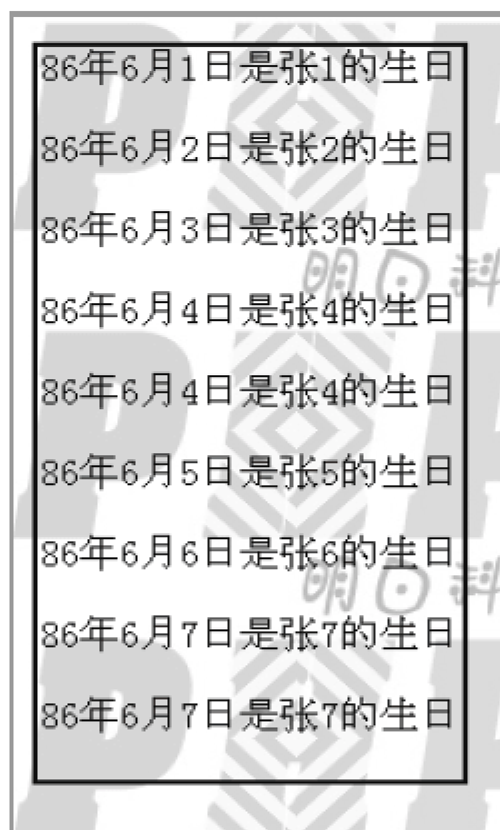


图 2.35　员工生日列表

关键技术

while 循环是 PHP 中最简单的循环控制语句。while()循环语句根据某一条件进行判断，决定是否执行循环，其语法格式是：

```
while (expr){
statement
}
```

当表达式 expr 的值为 true 时，将执行 statement 语句，执行结束后，再返回到 expr 表达式继续进行判断。直到表达式的值为 false 才跳出循环，执行大括号后面的语句。

设计过程

创建 index.php 文件。首先，利用数组函数将数据信息存储在数组变量中。然后，定义循环变量的初始值为 0，最大值不超过数组长度，并在循环内部做自增运算。最后，利用 while 循环输出结果，其代码如下：

```
<?php
	$a = array("86 年 6 月 1 日是张 1 的生日","86 年 6 月 2 日是张 2 的生日","86 年 6 月 3 日是张 3 的生日","86 年 6 月 4 日是张 4 的生日","86 年 6 月 4 日是张 4 的生日","86 年 6 月 5 日是张 5 的生日","86 年 6 月 6 日是张 6 的生日","86 年 6 月 7 日是张 7 的生日","86 年 6 月 7 日是张 7 的生日");	//定义数组
$b = 0;	//给变量赋初始值
	while($b < count($a)){	//根据数组长度循环输出
		echo $a[$b]."<br><br>";	//输出数组元素
		$b++;
	}
?>
```

秘笈心法

心法领悟 064：介绍 while()语句的变体。

while()循环语句还有另一种表示形式，即 do...while，详细说明请参见实例 065。

实例 065	员工详细信息浏览 光盘位置：光盘\MR\02\065	高级 趣味指数：★★★★

实例说明

本实例通过 do…while 循环语句循环读取数据库中的数据，输出员工的详细信息，运行结果如图 2.36 所示。

姓名：　年龄：　出生日期：　所在地址：　QQ：

姓名：阳光　年龄：25　出生日期：1986-06-06　所在地址：吉林省长春市　QQ：284126974

姓名：郭靖　年龄：25　出生日期：0000-00-00　所在地址：宋代汴梁　QQ：xxxxxxxxx

姓名：黄蓉　年龄：25　出生日期：0000-00-00　所在地址：宋代汴梁　QQ：xxxxxxxxx

图 2.36　员工详细信息浏览

关键技术

while()语句还有另外一种表示形式，即 do…while。do…while 循环语句和 while 循环语句非常类似，只是 do…while 循环语句在循环底部检测循环表达式，而不是在循环的顶部进行检测。do…while 循环语句的语法格式如下：

```
do{
        statment
}while(expr);
```

设计过程

（1）创建 PHP 脚本文件。首先，利用 mysql_connect()函数连接 MySQL 数据库，并通过 mysql_query 函数定义页面的编码格式。然后，定义 SQL 查询语句并返回结果集。最后，通过 do…while 语句循环输出数据库中的数据，其代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111")                //连接数据库服务器
mysql_select_db("db_database02",$conn);                        //连接数据库
mysql_query("SET NAMES GBK");                                  //设置编码格式
$rs = mysql_query("select * from tb_while");                   //执行查询
do{
        echo "姓名： ".iconv('gbk','utf-8',$rst[1])."  年龄： ".iconv('gbk','utf-8',$rst[2])."  出生日期：
".iconv('gbk','utf-8',$rst[3])."  所在地址： ".iconv('gbk','utf-8',$rst[4])."  QQ： ".iconv('gbk','utf-8',$rst[5])."<br><br>";
}while($rst = mysql_fetch_array($rs));
?>
```

（2）将该文件存储于\MR\02\065\文件夹中，命名为 index.php。运行结果如图 2.36 所示。

秘笈心法

心法领悟 065：使用 do…while 语句的注意事项。

在使用 do…while 语句之前，要考虑程序是否有必要在判断条件之前运行一次，如果没有必要，尽量不要使用该语句，否则可能会出现意外的输出结果。

实例 066　员工信息的批量删除

光盘位置：光盘\MR\02\066

高级　趣味指数：★★★★

实例说明

在操作数据库的过程中可能会出现很多无用的冗余信息，可以使用 while()循环语句将其循环删除。本实例通过 while()循环语句实现员工信息的批量删除，运行结果如图 2.37 所示。

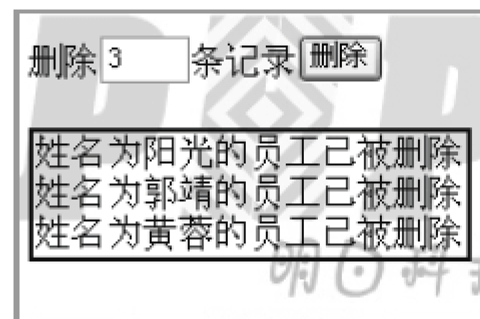


图 2.37　员工信息的批量删除

关键技术

本实例的关键点是 while()循环语句的灵活运用。根据提交的参数值，执行 while()循环语句，在循环体中执行删除操作，完成数据的循环删除操作。

设计过程

（1）创建 PHP 脚本文件。首先，连接 MySQL 数据库服务器，选择 MySQL 中名称为 db_while 的数据库，并设置页面的编码格式。然后，定义循环变量$a 和$b，将第 1 层 while 循环中的循环条件设置为$b<$a，然后，再次利用 while 语句循环输出数据表中的数据，并执行数据的删除操作，其代码如下：

```
<?php
if($_POST[sub]){                                                    //通过 POST 方式获取参数
        $conn = mysql_connect("localhost","root","111");            //连接 mysql 数据库
        mysql_select_db("db_while",$conn);                          //连接数据库
        mysql_query("SET NAMES GBK");                               //定义编码格式
        $a = $_POST[te];                                            //接收文本框参数
        $b = 0;                                                     //定义变量
            while($b < $a){                                         //while 循环
                $rs = mysql_query("select * from tb_while");        //执行查询操作
                while($rst = mysql_fetch_array($rs)){               //将查询结果保存在数组中
                    $sql = "delete * from tb_while where id = $b";  //sql 语句
                    mysql_query($sql);                              //执行删除操作
                    echo "姓名为".$rst[name]."的员工已被删除<br>";  //输出被删除的员工姓名
                    $b++;
                }
            }
    }
?>
```

（2）将该文件存储于\MR\02\066 文件夹下，命名为 index.php。运行结果如图 2.37 所示。

秘笈心法

心法领悟 066：使用 while()语句嵌套的注意事项。

while()循环语句允许语句嵌套，但在嵌套的过程中要注意层次结构的清晰。

实例 067　表格的动态创建

光盘位置：光盘\MR\02\067

高级　趣味指数：★★★★

实例说明

很多时候 PHP 编码需要和 HTML 编码混编，例如，输出表格打印 html 标签等。本实例应用两层 for()语句循环制作动态表格，运行结果如图 2.38 所示。

0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7
0	1	2	3	4	5	6	7

图 2.38　表格的动态创建

关键技术

for 循环是 PHP 中最复杂的循环结构。for 循环语句能够按照已知的循环次数进行循环操作，主要应用于多条件情况下的循环操作，不适合在单一条件下使用。这一点从该语句的语法中就可以看出，其条件的表达式有 3 个，语法格式为：

```
for (expr1; expr2; expr3){
statement;
}
```

其中，expr1 为变量初始赋值；expr2 为循环条件，即在每次循环开始前求值。如果其值为真，则执行 statement，否则，跳出循环，继续往下执行；expr3 为变量递增或递减，即每次循环后被执行。

设计过程

（1）创建 PHP 脚本文件。首先，定义 for 循环中的循环变量$a。设定$a 的初始值为 0，最大值为 4，并且每执行一次循环就做一次自增运算。然后，利用循环嵌套的知识定义内层循环，内层循环的循环变量$b 的初始值为 0，最大值为 7，并且同样做自增运算。最后，利用 echo 语句输出信息，其代码如下：

```
<?php
    for($a = 0;$a < 5;$a++){                                    //1 层 for 循环
        echo "<table border='1'bordercolor='#000099' bordercolorlight='#000099' cellspacing='0'>";
        echo "<tr>";                                            //表格标签
        for($b = 0;$b < 8;$b++){                                //2 层 for 循环
            echo "<td>";                                        //表格标签
            echo $b;                                            //输出数据
            echo "</td>";
        }
      echo "</tr>";
      echo "</table>";
        }
?>
```

（2）将该文件存储于\MR\02\067 文件夹下，命名为 index.php。运行结果如图 2.38 所示。

秘笈心法

心法领悟 067：for()循环语句。

for()循环语句在 PHP 循环结构中比较常用，结构也比较简单。所以一定要灵活掌握此循环语句。在本实例中，虽然其定义变量的最大值为 4 和 7，但是，由于在 for 循环中，其条件表达式的值是从等于 0 开始，所以循环语句分别执行了 5 次和 8 次。

实例 068	SESSION 购物车中数据的读取 光盘位置：光盘\MR\02\068	高级 趣味指数：★★★★

实例说明

购物车是电子商务系统中的常见模块，SESSION 作为一种会话机制作用不可小觑。本实例综合运用 while()语句、switch()语句、if()语句和 for ()语句实现一个简单的 SESSION 购物车功能，运行结果如图 2.39 所示。

1
山地自行车
1件
600元
30元（自理）
金牌用户VIP享受8.8折优惠!
2
液晶电脑显示器
1件
900元
30元（自理）
金牌用户VIP享受8.8折优惠!
3
液晶电脑显示器
1件
900元
30元（自理）
金牌用户VIP享受8.8折优惠!

图 2.39　SESSION 购物车中数据的读取

关键技术

（1）通过 switch()语句根据地址栏中传递的参数值，实现在不同商品页面之间的跳转。
（2）通过 if()语句对 form()表单提交的值进行判断，进而执行不同的操作。
（3）通过 while()语句和 for()语句完成购物车中购买商品的输出。

设计过程

（1）创建 index.php 文件，通过 switch()语句根据地址栏中传递的参数值，实现在不同商品页面之间的跳转，其代码如下：

```
<?php
	switch($_GET[link]){
		case "电脑";
		include_once("in_1.php");
		break;
		case "家庭影院";
		include_once("in_2.php");
		break;
		case "全自动洗衣机";
		include_once("in_3.php");
		break;
		case "数码摄像机";
		include_once("in_4.php");
		break;
		case "山地自行车";
		include_once("in_5.php");
		break;
		default:
		include_once("in_image.php");
	}
?>
```

（2）创建 config.php 文件。首先，编辑 form 表单，执行不同的操作。然后，应用 if()语句根据 form 表单提交的值进行判断，进而执行不同的操作。最后，通过 while()和 for()语句循环输出购物车中存储的商品，其关键代码如下：

```
	<form action="" method="post">
		我要购买<input type="text" size="2" value ="1" name="num">件
	   <input type="submit" name="sub" value="放入购物车" style="background-image:url(pic/button_2.jpg);color:red">
		<input type="submit" name="sub1" value="结账" style="background-image:url(pic/button_3.jpg);color:red">
		<input type="submit" name="sub2" value="首页" style="background-image:url(pic/button_1.jpg);color:red">
	</form>
<?php
		if($_POST[sub2]){
			echo "<script>location.href='index.php'</script>";
		}
		if($_POST[sub]){
			$str = explode("#",$_SESSION[id]);
			include("class/mysql.php");
			$sql = "insert into tb_shop(id,name,number,money,moneyother,other)values('','$str[0]','$str[1]','$str[2]','$str[3]','$str[4]')";
			if(mysql_query($sql)){
				mysql_close($conn);
				echo "<script>alert('已经放入购物车')</script>";
			}
		}
	if($_POST[sub1] || $_GET['sub1']){
		require_once("class/mysql.php");
		$sq = "select * from tb_shop";
		$rs = mysql_query($sq);
?>
		<form action ="config/table.php" method="post">
			<input type="submit" name="sub3" value="删除" style="background-image:url(pic/button_3.jpg);color:red">
```

```
                </form>
 <?php
        while($rst = mysql_fetch_row($rs))
                for($a = 0;$a < 6;$a++){
?>
                                <tr><td><?php echo $rst[$a]?></td></tr>
<?php
                }
?>
        </table>
        <a href="config/money.php"><img src="pic/bu.jpg"></a>
<?php
        }
?>
```

（3）创建 table.php 文件，执行删除指定商品的操作，其关键代码如下：

```
<?php
require_once("../class/mysql.php");                                 //包含数据库连接文件
                $sq = "select * from tb_shop";                      //定义 SQL 查询语句
                $rs = mysql_query($sq);                             //执行查询操作
?>
        <table>
<?php
                while($rst = mysql_fetch_row($rs)){                 //循环输出商品信息
?>
                <tr><td>商品编码：<?php echo $rst[0];?></td><td>商品名称：<?php echo $rst[1];?></td></tr>
                <tr><td>购买数量：<?php echo $rst[2];?><hr></td><td>金额总计：<?php echo $rst[3];?><hr></td></tr>
                <!--tr><td>商品运费：<?php echo $rst[4];?><hr></td><td>其他声明：<?php echo $rst[5];?><hr></td></tr-->
<?php
                }
?>
        </td>
        </tr></table></td></tr></table>
<?php
if($_POST[sub4]){
        $sqlser = "delete from tb_shop where id = '$_POST[tt]'";    //定义 SQL 语句
        if(mysql_query($sqlser)){                                   //执行删除操作
                echo "<script>alert('商品删除成功')</script>";
                echo "<script>window.location.href='table.php';</script>";
        }
}
if($_POST[sub5]){
        echo "<script>window.location.href='../index.php?link=电脑&sub1=1';</script>";
}
?>
```

（4）编辑 mysql.php 文件，实现与数据库的连接。

秘笈心法

心法领悟 068：知识对接。

购物车的知识相对较复杂，在本书第 5 章的实例 266 中将对购物车进行详细讲解。

实例 069	员工信息的管理 光盘位置：光盘\MR\02\069	高级 趣味指数：★★★★

实例说明

员工信息一般存在于人事信息平台的后台系统中，对员工信息的管理本身可以作为一个典型的模块被使用。本实例通过 for ()循环语句实现对员工数据的读取、修改和删除操作，运行结果如图 2.40~图 2.42 所示。

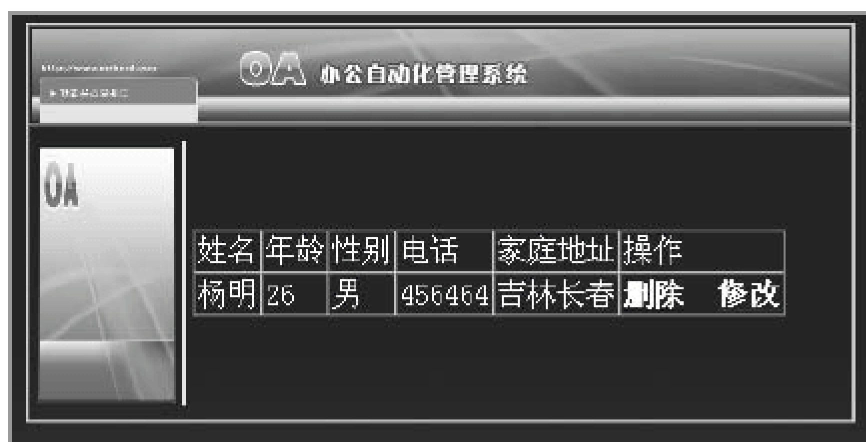


图 2.40　员工数据的读取



图 2.41　员工数据的删除

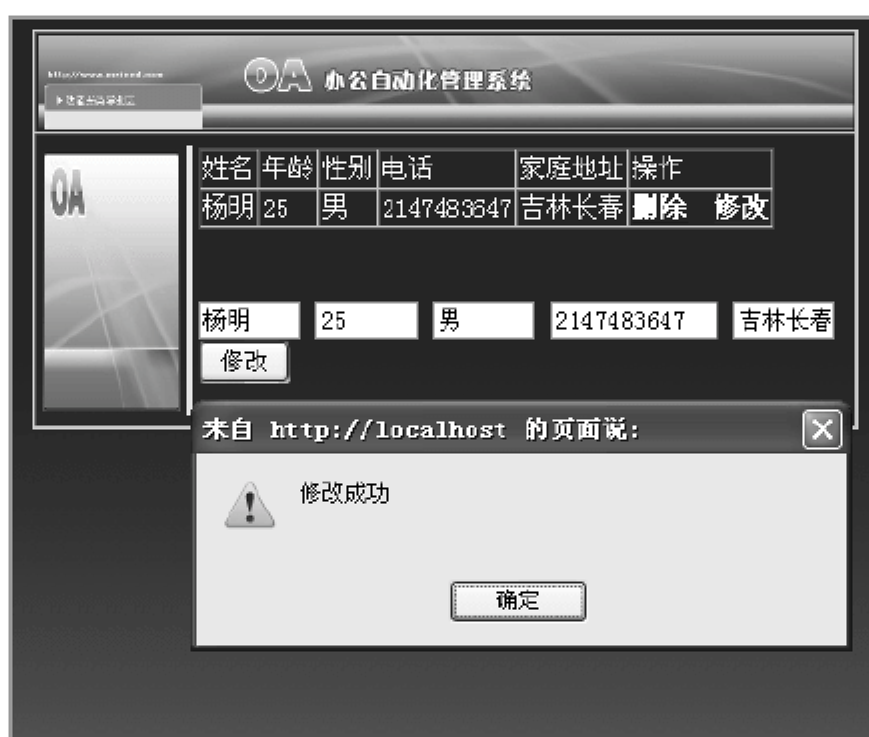


图 2.42　员工数据的修改

■ 关键技术

本实例通过 for()循环语句读取查询结果集中的数据，完成数据的循环输出，并且创建删除和修改超链接，应用 if()语句对超链接传递的参数值进行判断，实现对数据的删除和修改操作。有关 for()循环语句的详细说明，请参见实例 067，有关 if()语句的详细讲解请参考实例 059。

■ 设计过程

创建 index.php 文件。首先，连接 MySQL 数据库，执行查询操作，取得查询结果集。然后，通过 for()语句循环输出结果集中的数据。最后，通过 if()语句对超链接的参数值进行判断，进而执行删除和修改操作，其关键代码如下：

```
<?php
    $link = mysql_pconnect("localhost", "root", "111");                    //连接 mysql
    mysql_select_db("db_database02",$link);                                 //连接数据库
    mysql_query("SET NAMES GBK");                                           //设置编码格式
    $query = "SELECT * FROM tb_for_1";                                      //SQL 语句
    $result = mysql_query($query);                                          //执行查询操作
    for ($i = mysql_num_rows($result) - 1; $i >= 0; $i--) {                 //for 循环
        if (!mysql_data_seek($result, $i)) {                                //如果指针不向下移动
            echo "Cannot seek to row $i: " . mysql_error() . "\n";
            continue;
        }

        if(!($row = mysql_fetch_object($result)))                           //如果结果集不存在
            continue;

        echo "<tr><td>$row->name</td><td>$row->age</td><td>$row->sex</td><td>$row->tel</td><td>$row->address</td>
      <td><a href='1.php?id=$row->id&delete=1'><b>删除</b></a>  <a href='1.php?id=$row->id&update=1'><b>修改
</b></a></td></tr>";
```

```
    }
    mysql_free_result($result);                                    //释放结果集
if($_GET[delete] == 1){                                            //如果地址栏参数 delete 等于 1.
            $sql = "delete from tb_for_1 where id = '$_GET[id]'";  //SQL 语句
            if(mysql_query($sql)){                                 //执行删除操作
                echo "<script>alert('删除成功');location.href='1.php'</script>";   //提示框信息
            }
}
if($_GET[update] == 1){                                            //如果地址栏参数 update 等于 1.
            $sql = "select * from tb_for_1 where id ='$_GET[id]'"; //SQL 语句
            $rs = mysql_query($sql);                               //执行查询语句
            $rst = mysql_fetch_array($rs);                         //取得结果集
?>
</table><br><br>
      <form action="" method="post">                               //表单
        //使文本框初始值为数据库中结果
        <input name="name" type="text" value="<?php echo $rst[name];?>" size="6">
        <input name="age" type="text" value="<?php echo $rst[age];?>" size="6">
        <input name="sex" type="text" value="<?php echo $rst[sex];?>" size="6">
        <input name="tel" type="text" value="<?php echo $rst[tel];?>" size="11">
        <input name="address" type="text" value="<?php echo $rst[address];?>" size="6">
        <input type="submit" name="sub1" value="修改">
       </form></td></tr></table></td></tr></table>
<?php
            if($_POST[sub1]){                                      //如果按钮被单击
                $sql = "update tb_for_1 set name = '$_POST[name]',age = '$_POST[age]',sex = '$_POST[sex]',tel = '$_POST[tel]',address =
'$_POST[address]' where id =   '$_GET[id]'";//sql 语句

                if(mysql_query($sql)){                             //执行更新操作
                    echo "<script>alert('修改成功');location.href='1.php'</script>";   //提示框信息
                }
            }
}
?>
```

秘笈心法

心法领悟 069：使用地址栏传递参数存在风险。

通过地址栏传递参数是存在风险的，一般情况下应使用 POST 方式传递参数。如果一定要使用 GET 方式传递参数，最好应用 urlencode()函数对参数进行编码。

实例 070 网页版九九乘法表

光盘位置：光盘\MR\02\070

高级

趣味指数：★★★★

实例说明

在编程世界里，算法是一门独立于语法、函数之外的知识。它着重考验用户的逻辑思维能力并且与数学知识息息相关。本实例通过 for()循环语句实现网页版九九乘法表，运行结果如图 2.43 所示。

1 * 1 =1								
2 * 1 =2	2 * 2 =4							
3 * 1 =3	3 * 2 =6	3 * 3 =9						
4 * 1 =4	4 * 2 =8	4 * 3 =12	4 * 4 =16					
5 * 1 =5	5 * 2 =10	5 * 3 =15	5 * 4 =20	5 * 5 =25				
6 * 1 =6	6 * 2 =12	6 * 3 =18	6 * 4 =24	6 * 5 =30	6 * 6 =36			
7 * 1 =7	7 * 2 =14	7 * 3 =21	7 * 4 =28	7 * 5 =35	7 * 6 =42	7 * 7 =49		
8 * 1 =8	8 * 2 =16	8 * 3 =24	8 * 4 =32	8 * 5 =40	8 * 6 =48	8 * 7 =56	8 * 8 =64	
9 * 1 =9	9 * 2 =18	9 * 3 =27	9 * 4 =36	9 * 5 =45	9 * 6 =54	9 * 7 =63	9 * 8 =72	9 * 9 =81

图 2.43　网页版九九乘法表

关键技术

本实例的关键点是利用 for 循环中的第 2 层循环控制第 1 层循环，以实现指定数据的输出。

设计过程

（1）创建 PHP 脚本文件，定义 1 层循环中的循环变量为$a，设置$a 的初始值为 1，最大值为 9，并且每执行一次循环就做一次自增运算。然后，定义内层循环中的循环变量为$b，设置$b 的初始值为 1，最大值小于$a，并且做自增运算。最后，将 2 层循环控制的数据输出到页面，其代码如下：

```
<?php
	for($a = 1;$a < 10;$a++){					//1 层循环
		echo "<tr>";
		for($b = 1;$b <= $a;$b++){				//2 层循环控制 1 层循环输出
			echo "<td>";
			echo "$a * $b =".$a*$b;			//输出结果
			echo"</td>";
		}
		echo "</tr>";
	}
?>
```

（2）将该文件存储于\MR\02\070 文件夹中，命名为 index.php。运行结果如图 2.43 所示。

秘笈心法

心法领悟 070：形成良好的算法思维。

算法不是一朝一夕就可以掌握的，是一个慢慢了解、形成思想的过程。

实例 071	读取数组购物车中的数据 光盘位置：光盘\MR\02\071	高级 趣味指数：★★★★

实例说明

用户在进行开发时，没有必要将小型的数据与数据库进行交互，可以直接将其存入数组，这样不仅可以节约开发时间，还可节省服务器资源。本实例通过对数组函数的相关操作实现读取数组购物车中的数据，运行结果如图 2.44 所示。



图 2.44　实现读取

关键技术

本实例的关键点是运用 foreach()语句遍历数组，取得 key 值和对应的 value 值。foearch 循环在 PHP 4.0 中擅长处理数组，是遍历数组的一种简单方法；在 PHP 5.0 中，新增加了对对象的支持。该语句的语法格式为：

```
foreach (array_expression as $value)
statement
```

或

```
foreach (array  expression as $key => $value)
    statement
```

forearch()语句将遍历数组 array_expression，每次循环时，将当前数组中的值赋给$value（或是$key 和$value），同时，数组指针向后移动，直到遍历结束。当使用 foreach()语句时，数组指针将自动被重置，所以不需要手动设置指针位置。

设计过程

（1）创建 PHP 脚本文件。首先，定义字符串变量，并且在变量中插入“#”作为字符串分割点。然后，利用 GET 方式接收地址栏传递的参数。当$_GET[push]的值等于 1 时，输出 JavaScript 成功提示。当$_GET[pop]的值等于 1 时，利用字符串拆分函数将字符串变量以#分割，并分别保存到数组之中。最后，利用 foreach()语句将字符串数组中的 value 值输出到页面中，其代码如下：

```
<?php
    $str = "宝贝：液晶电脑显示器#一口价：900 元#运费：30 元（自理）";        //定义字符串
        if($_GET[push] == 1){                                              //通过 GET 方式取得参数
            echo "<script>alert('成功放入购物车');</script>";               //提示消息
        }
        if($_GET[pop] == 1){                                               //通过 GET 方式取得参数
            $strexplode = explode("#",$str);                               //分割字符串
            foreach($strexplode as $key => $value){                        //遍历数组
?>
            <tr><td><?php echo $value;?></td></tr>                         //输出结果
<?php
            }
        }
?>
```

（2）将该文件存储于\MR\02\071 文件夹中，命名为 index.php。运行结果如图 2.44 所示。

秘笈心法

心法领悟 071：对于 foreach()语句的说明。

foreach()语句仅能用于数组，当试图将其用于其他数据类型或者一个未初始化的变量时会产生错误。foreach()语句不支持用“@”来禁止错误信息。

实例 072	图像验证码的生成 光盘位置：光盘\MR\02\072	高级 趣味指数：★★★★

实例说明

用户注册后，系统为了保存用户的一些基本信息，包括用户名、密码等相关信息，以及跟踪和定位不同用户，从而更好地服务于用户，同时，为了提高站点的安全性，避免由于网速慢造成用户注册信息的重复提交，往往会在用户注册表中增加验证码功能。本实例通过 for 循环实现图像验证码的生成，运行结果如图 2.45 所示。

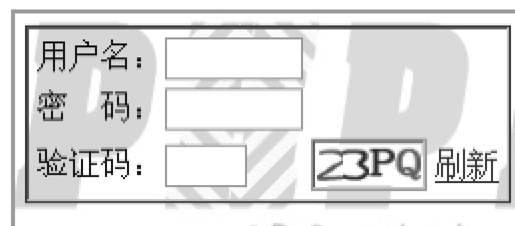


图 2.45　图像验证码的生成

关键技术

本实例的关键点是 for()循环语句和 rand()函数的灵活运用。通过 for()循环输出一个包含 4 个图像的验证码，验证码的取值由随机函数 rand()生成。

rand()函数，可产生一个随机整数，其语法如下：

```
int rand ( [int min, int max] )
```

如果没有提供可选参数 min 和 max，rand()返回 0 到 RAND_MAX 之间的伪随机整数。例如想得到 2~8（包括 2 和 8）之间的随机数，则写为 rand(2, 8)。

设计过程

（1）创建 PHP 脚本文件。首先，利用数组函数定义数组并分别将图片的地址信息存入到数组之中。然后，在 for 循环的循环体内，利用随机函数取得 0~3 中的随机数字并保存在变量$rand 中。最后，利用拼接字符串的方法输出图片，其代码如下：

```
<?php
	$array = array("pic/22.jpg","pic/3.jpg","pic/4.jpg","pic/5.jpg");		//定义数组
	for($a = 0;$a < 4;$a++){							//for 循环
		$rand = rand(0,3);							//0~3 的随机数字
		echo "<img src = '$array[$rand]'>";					//输出图像
	}
?>
```

（2）将该文件存储于\MR\02\072 文件夹中，命名为 index.php。运行结果如图 2.45 所示。

秘笈心法

心法领悟 072：图像验证码的刷新。

在本实例中，通过重新链接到首页来实现验证码的刷新操作。

2.6　跳 转 语 句

跳转控制语句包括 break 和 continue 两个。其中 break 语句可以终止当前的循环；而 continue 语句只能终止本次循环，并进入到下一次循环中。

实例 073	控制页面中表情图的输出 光盘位置：光盘\MR\02\073	高级 趣味指数：★★★★

实例说明

break 语句的含义是结束当前 for、foreach、while、do...while 或者 switch 结构的执行。本实例通过 break()语句实现控制页面中表情图的输出，运行结果如图 2.46 所示。



图 2.46　控制页面中表情图的输出

关键技术

使用 break 关键字可以终止当前的循环，包括 while、do...while、for、foreach 和 switch 在内的所有控制语句，

其在 for()语句中的应用示例如下：

```
for($i=1;$i<=4;$i++){                          //应用 for 循环控制语句输出表情头像
 if($i==4){                                    //判断变量是否等于 4
        break;                                 //如果等于 4，使用 break 语句跳转循环
}
```

本实例将 break 语句与 switch()语句联合使用，控制页面中表情图的输出。

设计过程

（1）创建 PHP 脚本文件，当单击“确定”按钮时，利用 switch()语句选择输出的图片并通过 break 语句跳出当前 switch()语句，其代码如下：

```
<?php
        if($_GET[sub]){                                //通过 GET 方式接收参数
                switch($_GET['check']){                //switch()语句
                        case "1";                      //如果值为 1
                        echo "<img src='image/1.gif'>";   //输出图像 1
                        break;                         //跳出语句
                        case "2";                      //如果值为 2
                        echo "<img src='image/2.gif'>";   //输出图像 2
                        break;                         //跳出语句
                        case "3";                      //如果值是 3
                        echo "<img src='image/3.gif'>";   //输出图像 3
                        break;                         //跳出语句
                        case "4";                      //如果值为 4
                        echo "<img src='image/4.gif'>";   //输出图像 4
                        break;                         //跳出循环
                }
        }
?>
```

（2）将该文件存储于\MR\02\073 文件夹中，命名为 index.php。运行结果如图 2.46 所示。

秘笈心法

心法领悟 073：break 语句的妙用。

break 语句可以接受一个可选的数字参数来决定跳出几重循环。

实例 074　控制页面中数据的输出数量

光盘位置：光盘\MR\02\074

高级

趣味指数：★★★★

实例说明

break 语句可以指定跳出循环的层数，一般被用在控制条件方面。本实例通过 break 语句实现控制页面中数据的输出数量，运行结果如图 2.47 所示。

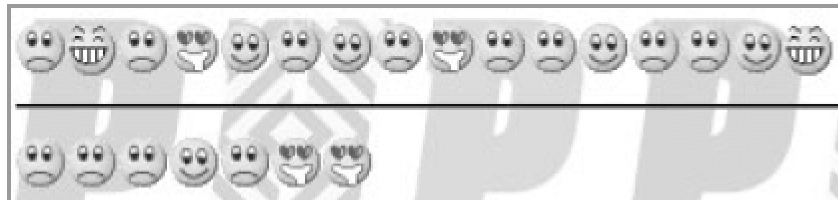


图 2.47　控制页面中数据的输出数量

关键技术

本实例运用 break 语句跳出指定的几重循环，其语法格式如下：

```
break n;
```

设计过程

（1）创建 PHP 脚本文件。首先，利用多重 for 循环嵌套输出全部图片。然后，在多重嵌套中定义 if()条件语句，当变量$rand 中的随机数值为 2 时，跳出 3 重循环。否则，正常输出图片，其代码如下：

```
<?php
    for($a = 0;$a < 2;$a++){                          //1 层循环
        for($b = 0;$b < 2;$b++){                      //2 层循环
            for($c = 0;$c < 4;$c++){                  //3 层循环
                $rand = rand(1,4);                    //1~4 随机数字
                echo "<img src='image/$rand.gif'>";   //输出图像
            }
        }
    }
    echo "<hr style='color:blue;'>";
    for($a = 0;$a < 2;$a++){                          //1 层循环
        for($b = 0;$b < 2;$b++){                      //2 层循环
            for($c = 0;$c < 4;$c++){                  //3 层循环
                $rand = rand(1,4);                    //1~4 随机数字
                if($rand != 2){                       //条件语句
                    echo "<img src='image/$rand.gif'>";  //输出图像
                }else{
                    break 3;                          //结束 3 个层级的循环
                }
            }
        }
    }
?>
```

（2）将该文件存储于\MR\02\074 文件夹中，命名为 index.php。运行结果如图 2.47 所示。

秘笈心法

心法领悟 074：灵活运用跳转语句。

跳转语句一定要灵活运用。常用的跳转语句包括 break 语句和 continue 语句，continue 语句将在后面的实例中做详细介绍。

实例 075　动态改变页面中单元格的背景颜色

光盘位置：光盘\MR\02\075

高级

趣味指数：★★★★

实例说明

在循环结构中，有些数据是不需要的，想要跳过这些代码向下执行，这时候就需要用到 continue 语句。本实例通过 continue 语句实现动态改变页面中单元格的背景颜色，运行结果如图 2.48 所示。



图 2.48　动态改变页面中单元格的背景颜色

关键技术

在程序执行 break 后，程序将跳出循环执行流程，而开始继续执行循环体的后续语句。continue 跳转语句的作用没有 break 那么强大，continue 只能终止本次循环，而进入到下一次循环中。在执行 continue 语句后，程序

将结束本次循环的执行，并开始下一轮循环的执行操作。continue 跳转语句的流程控制图如图 2.49 所示。

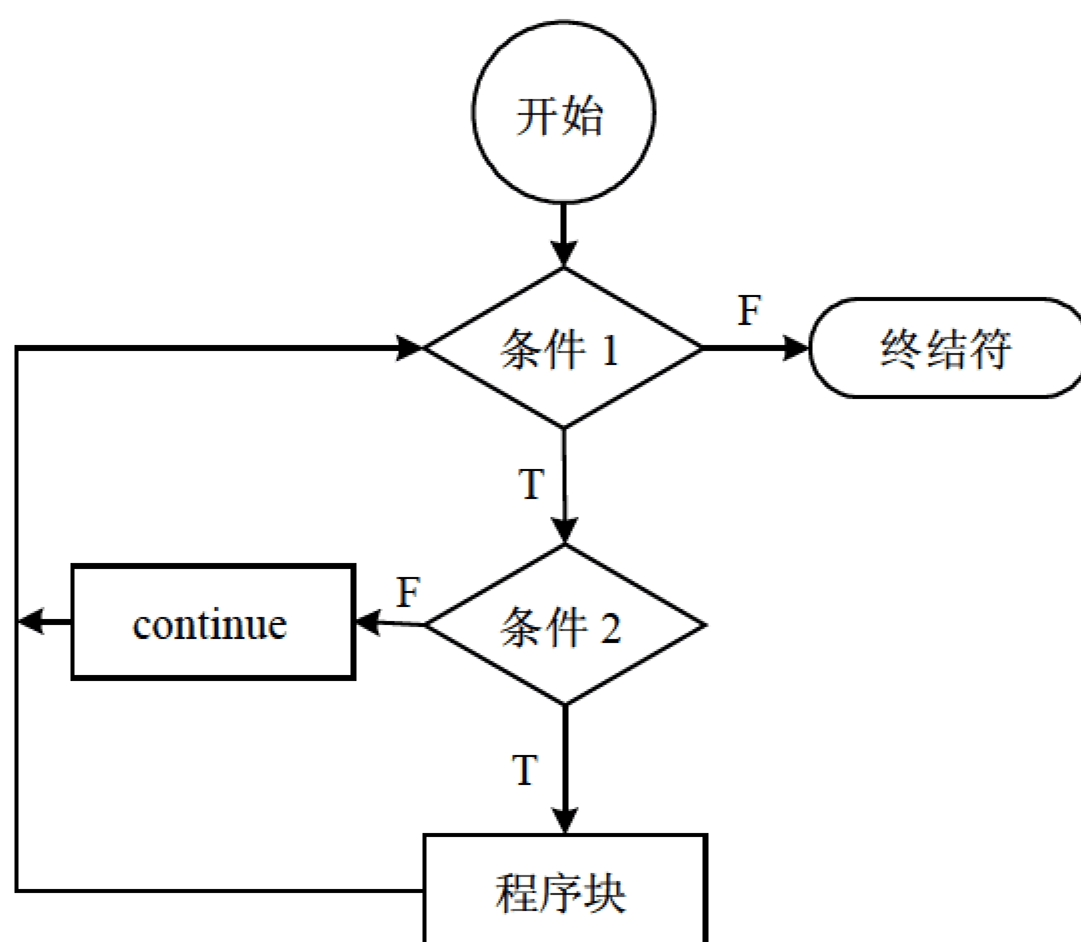


图 2.49　continue 流程控制图

设计过程

（1）创建 PHP 脚本文件。首先，利用 for 循环定义循环变量的初始值为 0，最大值为 9，并且每执行完一次循环，循环变量做一次自增运算。然后，利用 if()语句定义条件，当循环变量小于等于 5 时，通过 dechex 函数获取 6 个十六进制随机数，否则终止本次循环。最后，将取得的 6 个十六进制数添加到<table>标签的 bgcolor 属性中，实现动态变色，其代码如下：

```
<?php
	for($b = 0;$b < 10;$b++){				//循环语句
		if($b <= 5){					//条件语句
			$rand.= dechex(rand(0,15));		//取得 6 个十六进制数
		}else{
			continue;				//不满足条件，重新执行循环
		}
	}
?>
<table height="55" bgcolor="#<?php echo $rand;?>">//设置背景颜色
  <tr><td width="156"></td>
</tr></table>
```

（2）将该文件存储于\MR\02\075 文件夹中，命名为 index.php。运行结果如图 2.48 所示。

秘笈心法

心法领悟 075：continue 语句的妙用。

continue 语句也可以指定跳转层级数。

实例 076	屏蔽偶数次的数据输出 光盘位置：光盘\MR\02\076	高级 趣味指数：★★★★

实例说明

数学中有这样一道题：取得从 0~10 所有偶数或者奇数的和。如果想把这道题用 PHP 代码表示出来，则需

要使用跳转语句 continue 来完成。本实例通过跳转语句 continue 实现屏蔽偶数次的数据输出，运行结果如图 2.50 所示。

1 * 1 =1	1 * 3 =3	1 * 5 =5	1 * 7 =7	1 * 9 =9
3 * 1 =3	3 * 3 =9	3 * 5 =15	3 * 7 =21	3 * 9 =27
5 * 1 =5	5 * 3 =15	5 * 5 =25	5 * 7 =35	5 * 9 =45
7 * 1 =7	7 * 3 =21	7 * 5 =35	7 * 7 =49	7 * 9 =63
9 * 1 =9	9 * 3 =27	9 * 5 =45	9 * 7 =63	9 * 9 =81

图 2.50　屏蔽偶数次的数据输出

关键技术

本实例的关键点是 continue 语句的灵活运用，详细说明请参见实例 075。

设计过程

创建 PHP 脚本文件。首先通过 for()循环语句动态输出 HTML 的<tr>标签和数据，并且当循环变量为偶数时终止本次循环。然后，利用相同的方法动态输出 HTML 的<td>标签和数据。最后，在页面中输出如图 2.50 所示的表格，其代码如下：

```
<?php
	for($a = 1;$a < 10;$a++){				//1 层循环
		if($a % 2 == 0){					//如果为偶数则回到本循环开始位置
			continue;
		}
		echo "<tr>";
		for($b = 1;$b < 10;$b++){			//2 层循环
			if($b % 2 != 0){				//如果为偶数
				echo "<td>";
				echo "$a * $b =".$a*$b;		//输出结果
				echo "</td>";
			}else{
				continue;					//回到本循环的开始位置
			}
		}
		echo "</tr>";
	}
?>
```

秘笈心法

心法领悟 076：求余运算符。

利用“%”运算符可取得两个数的余数，本实例通过此运算符判断是否为偶数，如果想要判断是否为奇数只需将条件设定为不等于 1 即可。

实例 077　跳过数据输出中指定的记录

光盘位置：光盘\MR\02\077

高级

趣味指数：★★★★

实例说明

本实例通过 continue 语句实现跳过数据输出中指定的记录，运行结果如图 2.51 所示。

关键技术

本实例应用 continue 语句跳过指定的循环，输出指定循环以外的数据。



图 2.51　跳过数据输出中指定的记录

设计过程

（1）创建 PHP 脚本文件。首先，定义一个数组来存储数据。然后，定义下拉列表，其中下拉列表的值由 PHP 代码动态添加，当输入跳过记录的编号并单击“跳过”按钮时，利用 continue 语句，输出不包含指定数据的其他数据，其代码如下：

```
<?php
//定义数组
	$array = array("PHP 典型模块","PHP 开发实战宝典","JAVA 开发实战宝典","PHP 网络编程自学手册");
?>
//表单
<form action="" method="post">
<textarea name="te" cols="20" rows="6">0、<?php echo $array[0];?>		1、<?php echo $array[1];?>		2、<?php echo $array[2];?>
3、<?php echo $array[3];?>
</textarea><br>
<input type="text" name="text" value="输入要跳过记录的编号"size="20" onFocus="this.value=''">
<input type="submit" name="sub1" value="跳过">
</form>
<?php
	if($_POST[sub1]){					//通过 POST 方式传递参数
		for($a = 0;$a < count($array);$a++){		//循环数组
			if($a == $_POST[text]){			//当$a 等于文本框中数据时回到循环开始位置
				continue;
			}else{
				echo '《';
				echo $array[$a];		//输出数组元素
				echo '》';
			}
		}
	}
?>
```

（2）将该文件存储于\MR\02\077 文件夹中，命名为 index.php。运行结果如图 2.51 所示。

秘笈心法

心法领悟 077：在 HTML 中插入 PHP 代码。

PHP 代码可以插入到 html 页面的任何位置，也可以插入到 html 的任何一个可行性标签之中。

实例 078	执行指定次数的循环 光盘位置：光盘\MR\02\078	高级 趣味指数：★★★★

实例说明

本实例利用 break 语句实现执行指定次数的循环，运行结果如图 2.52 所示。

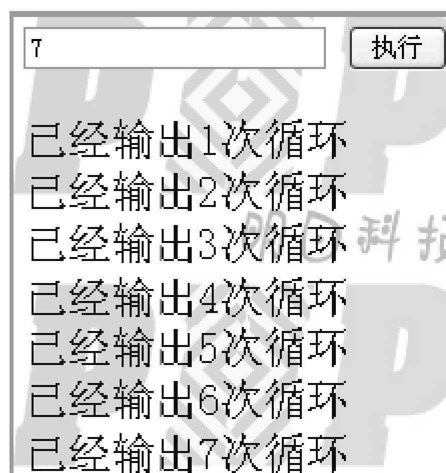


图 2.52　执行指定次数的循环

关键技术

本实例利用 break 语句可以跳出循环并终止循环的特点，控制循环执行的次数。

设计过程

（1）创建 PHP 脚本文件。当单击“执行”按钮时，首先，定义 for 循环语句，设定循环变量$a 的初始值为 0，并且每执行一次循环，$a 做一次自增运算。然后，通过 if()语句定义条件，如果$a 大于等于文本框中输入的数字时，跳出当前循环，否则输出字符串，其代码如下：

```
<?php
	if($_POST[sub1]){                                    //通过 POST 方式传递参数
		for($a = 0;$a < 10000000000000;$a++){             //循环
			if($a >= $_POST[text]){                       //接收文本框信息判断条件
				break;                                    //结束循环
			}else{                                        //否则循环输出
				echo "已经输出".($a+1)."次循环<br>";
			}
		}
	}
?>
```

（2）将该文件存储于\MR\02\078 文件夹下，命名为 index.php。运行结果如图 2.52 所示。

秘笈心法

心法领悟 078：流程控制语句。

流程控制语句包括条件控制语句、循环控制语句和跳转控制语句，是 PHP 的重要组成部分，也是 PHP 程序员必须重点掌握和理解的内容。

2.7　自定义函数

在开发过程中，经常需要多次重复某种操作或处理，如数据查询、字符操作等。如果每个模块都要重新输入一次代码，不仅加大程序员的工作量和开发时间，而且对于代码的后期维护及运算效果也有很大影响。使用自定义函数就是解决该问题的最有效的方法。

实例 079	自定义函数截取中文字符串 光盘位置：光盘\MR\02\079	高级 趣味指数：★★★★

实例说明

截取字符串在程序设计中是经常出现的，例如，截取增删改查 SQL 语句的前 6 个字符进行判断，以便进行

增删改查等操作。本实例应用 substr()函数实现用自定义函数截取中文字符串，运行结果如图 2.53 所示。

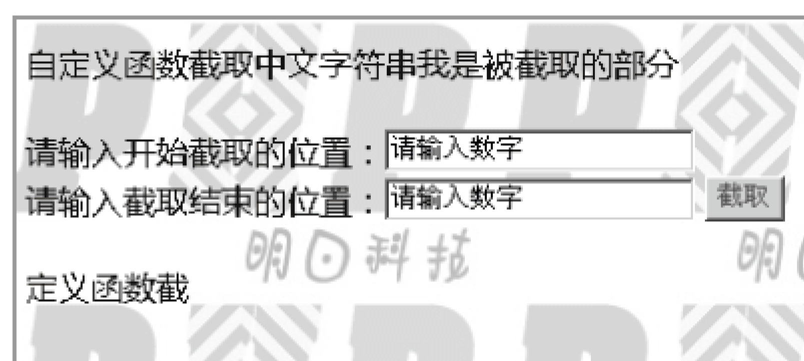


图 2.53　自定义函数截取中文字符串

关键技术

substr()函数用于从字符串中按照指定位置截取一定长度的字符，通过该函数可以获取某个固定格式字符串中的一部分。如果使用一个正数作为子串的起点来调用这个函数，将得到从起点到字符串结束的整个字符串；如果使用一个负数作为子串的起点来调用，将得到原字符串尾部的一个子串，字符个数等于给定负数的绝对值。Substr()函数的语法如下：

```
string substr ( string str, int start [, int length]);
```

参数 str 用来指定字符串对象；参数 start 用来指定开始截取字符串的位置，如果为负数，则从字符串的末尾开始截取；参数 length 为可选项，指定截取字符的个数，如果其为负数，则表示截取到倒数第 length 个字符。

设计过程

（1）创建 PHP 脚本文件，定义自定义函数，当单击“截取”按钮时，利用 substr()函数对指定字符串进行截取并输出截取信息，其代码如下：

```
<?php
	function str(){                                                      //自定义函数 str()
		if($_POST[sub]){                                                 //通过 POST 方式传递参数
			$a = "自定义函数截取中文字符串我是被截取的部分";                    //定义字符串变量
			$rs = substr($a,(2*$_POST['text']-2),(2*$_POST['text_1']));  //截取字符串
			echo $rs;                                                    //输出截取结果
		}
	}
str();                                                                   //自定义函数调用
?>
```

（2）将该文件存储于\MR\02\079\文件夹中，命名为 index.php。运行结果如图 2.53 所示。

秘笈心法

心法领悟 079：substr()函数的特点。

substr()函数支持从右侧取值，例如：

```
<?php
	$a ="abcdef";               //定义字符串
	$rs = substr($a,-2,1);      //从右侧取值
	echo $rs;                   //输出截取结果
?>
```

经 IE 浏览器运行，显示结果为“e”。

实例 080	公告标题的截取 光盘位置：光盘\MR\02\080	高级 趣味指数：★★★★

实例说明

本实例通过字符串截取函数和自定义函数实现公告标题的截取，运行结果如图 2.54 所示。

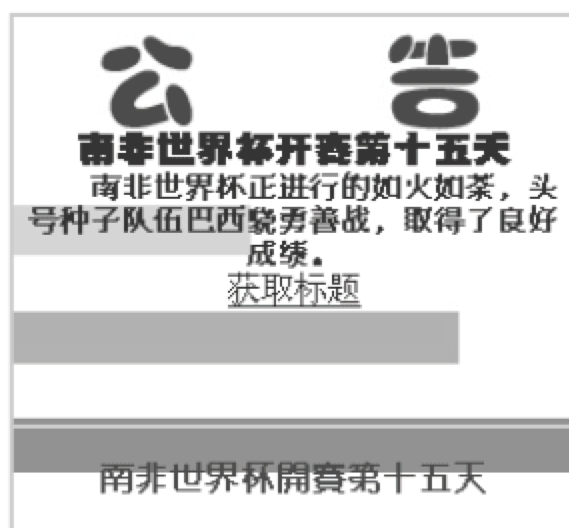


图 2.54　公告标题的截取

关键技术

自定义函数，顾名思义是由用户自行定义并完成指定功能的函数。自定义函数的产生是由于程序员在进行日常开发时，使用 PHP 预定义的函数并不能实现理想的功能。程序员迫切希望能够根据自己程序的实际情况自行定义完成指定功能的函数。创建函数的基本语法格式为：

```
function function_name ([$arg_1],[$arg_2], ... ,[$arg_n]){
    fun_body;
[return arg_n;]
}
```

参数说明：

function：声明自定义函数时必须使用的关键字。

function _name：创建函数的名称，是有效的 PHP 标识符。函数名称是唯一的，其命名遵守与变量命名相同的规则，只是不能以“$”开头。

arg_1…arg_n：是外界传递给函数的值，可有可无，也可以有多个，数量根据需要而定，各参数用逗号“,”分隔。参数的类型不必指定，在调用函数时只要是 PHP 支持的类型都可以使用。

fun_body：自定义函数的主体，是功能实现部分。

return：将调用的代码需要的值返回，并结束函数的运行。

在本实例中，定义自定义函数 str，对字符串进行截取操作，其语法如下：

```
function str($str){                              //自定义函数
        $a = strlen($str);                       //计算字符串长度
        $b = substr($str,0,$a);                  //截取字符串
        echo $b;                                 //输出
}
```

设计过程

（1）创建 PHP 脚本文件，定义数组，当单击“获取标题”按钮时，调用自定义函数 str 对数组中元素进行截取，并输出截取后的标题信息，其代码如下：

```
<?php
    $array = array("南非世界杯开赛第十五天","南非世界杯正进行得如火如荼，头号种子队伍巴西骁勇善战，取得了良好成绩。");
//定义数组
?>
<table border="1"   bordercolor="#CCCCCC"cellspacing="0" width="290" height="251" align="center"   background="2.jpg">
   <tr><td width="208" align="center">
   <b style="font-family:'文鼎霹雳体'; font-size:18px; color:#0033CC"><?php echo $array[0];?></b><br>
   <b style="font-family:'方正新报宋简体'; font-size:14px; color:#0033CC">      <?php echo
$array[1];?></b><br>
   <a href="index.php?sub=获取" style="float:center">获取标题</a>              //定义链接
   </td>
</tr><tr><td align="center"><b style="color:#FF0000; font-family:'方正细圆繁体'">
<?php
        if($_GET[sub] == "获取"){                                        //通过 GET 方式获取参数
```

```
            str($array[0]);                                         //调用自定义函数
        }
        function str($str){                                         //自定义函数
            $a = strlen($str);                                      //计算字符串长度
            $b = substr($str,0,$a);                                 //截取字符串
            echo $b;                                                //输出
        }
?>
```

（2）将该文件存储于\MR\02\080\文件夹下，命名为 index.php。运行结果如图 2.54 所示。

秘笈心法

心法领悟 080：自定义函数的应用注意事项。

自定义函数的名称是不区分大小写的，而常量和变量的名称区分大小写。

实例 081 论坛内容的简短输出

光盘位置：光盘\MR\02\081

高级

趣味指数：★★★★

实例说明

开发论坛程序很重要的一点是转换空格符和回车符，因为不是所有的论坛用户都知道 HTML 标签 和
的作用，大多数用户都喜欢用空格键和回车键对文本进行控制，这就需要程序员人为地做到统一。本实例通过自定义函数定义转换方法实现论坛内容的简短输出，运行结果如图 2.55 所示。

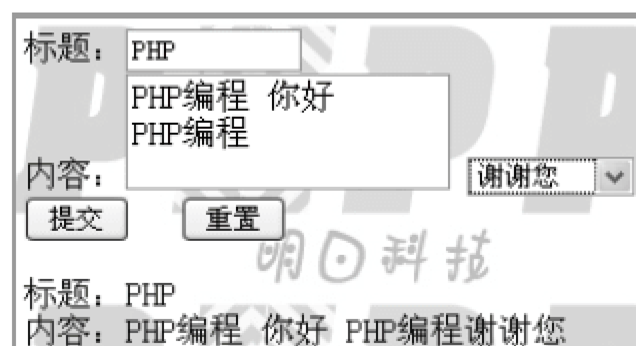


图 2.55　论坛内容的简短输出

关键技术

本实例通过自定义函数实现对字符串中空格等和回车符的转换输出，在自定义函数中应用的是 preg_replace()函数。自定义函数的语法如下：

```
function str($str){                                                 //自定义函数
            $str = preg_replace("/ /"," ",$str);               //替换空格
            $str = preg_replace("/cha(13)/","<br>",$str);           //替换回车符
            echo "内容： ".$str;                                     //输出
}
```

设计过程

（1）创建 PHP 脚本文件，通过自定义函数定义对 和回车进行替换，当单击“提交”按钮时，首先对文本框是否存在内容进行判断，如果文本框内容不为空，调用自定义函数并输出文本信息，其代码如下：

```
<?php
        function str($str){                                         //自定义函数
            $str = preg_replace("/ /"," ",$str);               //替换空格
            $str = preg_replace("/cha(13)/","<br>",$str);           //替换回车符
            echo "内容： ".$str;                                     //输出
        }
        if($_POST[sub]){                                            //通过 POST 方式传参
        //如果 text 为空或者文本域和下拉列表框同时为空
```

```
            if($_POST[text] == "" || (($_POST[te] == "") && ($_POST[check] == 1))){
                echo "<script>alert('内容不能为空');location.href='index.php'</script>";        //输出提示
            }else{
                    if($_POST[select] != '1'){                                              //当下拉列表值不等于 1 时
                        echo "标题：".$_POST[text]."<br>";                                 //输出内容和下拉列表内容
                        str($_POST[te].$_POST[check]);
                    }else{
                        echo "标题：".$_POST[text]."<br>";                                 //否则只输出文本域内容
                        str($_POST[te]);
                    }
            }
        }
?>
```

（2）将该文件存储于\MR\02\081\文件夹下，命名为 index.php。运行结果如图 2.55 所示。

秘笈心法

心法领悟 081：对于实例代码的重要说明。

上例代码中“$_POST[text] == "" || (($_POST[te] == "") && ($_POST[check] == 1))”是存在运算符优先级的，为了避免发生错误，笔者将先进行的运算人为加上小括号。

实例 082　自定义函数过滤字符串

高级

光盘位置：光盘\MR\02\082

趣味指数：★★★★

实例说明

用户在论坛上发表或者回复留言时可能会遇到这种情况，单击“发布”按钮时，系统提示使用了禁用词语，禁止发布。这是论坛的开发人员为了规范论坛而采用的过滤字符串的手段。本实例通过自定义函数实现过滤字符串，运行结果如图 2.56 所示。



图 2.56　自定义函数过滤字符串

关键技术

本实例编写自定义函数 str()，对字符串进行过滤，如果出现指定的关键词，则给出提示信息，并终止程序执行。自定义函数 str()的语法如下：

```
function str($str){
    $array = array('图书','明日科技','软件','编程词典','编程','词典');          //定义数组
    $repstr = implode($array);                                              //数组转换成字符串
    if(preg_match("/$str/",$repstr)){                                       //正则表达式验证字符串
        echo "<script>alert('您使用了禁用词语，请重新填写');location.href='index.php'</script>";
    }else{
        echo "内容为：".$str;                                                //输出数据
    }
}
```

设计过程

（1）创建 PHP 脚本文件。首先定义自定义函数，将禁用词语保存在数组中，并通过字符串函数拆分数组，利用正则表达式函数对禁用词语进行屏蔽。当单击“提交”按钮时，调用自定义函数对提交内容进行过滤，并输出提示信息，其代码如下：

```
<?php
function str($str){
	$array = array('图书','明日科技','软件','编程词典','编程','词典');		//定义数组
	$repstr = implode($array);								//数组转换成字符串
	if(preg_match("/$str/",$repstr)){							//正则表达式验证字符串
		echo "<script>alert('您使用了禁用词语，请重新填写');location.href='index.php'</script>";
	}else{
		echo "内容为：".$str;							//输出数据
	}
}
if($_POST[sub]){
	str($_POST[te]);									//调用自定义函数
}
?>
```

（2）将该文件存储于\MR\02\082\文件夹下，命名为 index.php。运行结果如图 2.56 所示。

秘笈心法

心法领悟 082：过滤程序。

本实例只是简单地介绍了字符串过滤的原理，真正的过滤程序由于数据量较大需要将数据存储在数据库中。如果读者感兴趣，可以将其与数据库中的数据结合使用。

2.8 字 符 串

在许多 Web 编程中，字符串总是会被大量地生成和处理。正确地使用和处理字符串，对于 PHP 程序员来说越来越重要。

实例 083	过滤论坛帖子中的空白和特殊字符 光盘位置：光盘\MR\02\083	高级 趣味指数：★★★★

实例说明

由于各个用户所使用的输入法不同或者个人习惯的关系，有可能不经意间多输入了空格或者特殊字符，造成网站内容的输出不美观。本实例通过 ltrim()和 rtrim()函数实现去除字符串的空白和特殊字符，运行结果如图 2.57 所示。

去除字符串空白和特殊字符前：(:@_@ 你好！PHP范例 @_@:)
去除字符串空白和特殊字符后： 你好！PHP范例

图 2.57　去除字符串空白和特殊字符

关键技术

（1）ltrim()函数

使用 ltrim()函数默认情况下去除左边空格，去除特殊字符的参数可选。

```
string ltrim ( string str [, string charlist]);
```

参数说明：

string str：源字符串信息。

string charlist：可选参数，特殊字符。

（2）rtrim()函数

使用 rtrim()函数默认情况下去除右边空格，去除特殊字符的参数可选。

```
string rtrim ( string str [, string charlist]);
```

参数说明：

string str：源字符串信息。

string charlist：可选参数，特殊字符。

设计过程

（1）创建 PHP 脚本文件，定义字符串变量$a，分别通过 ltrim()和 rtrim()函数去除两边的空格和特殊字符并显示结果，其代码如下：

```
<?php
    $a = "(:@_@   你好！PHP 范例   @_@:)";          //声明一个带空格和特殊字符的字符串变量
    $b = ltrim($a);                                 //去除变量左边的空格
    $c = ltrim($b,"(:@_@");                         //去除变量左边的特殊字符
    $d = rtrim($c);                                 //去除变量右边的空格
    $e = rtrim($d,"@_@:)");                         //去除变量右边的特殊字符
    echo "去除字符串空白和特殊字符前：".$a."<br>";
    echo "去除字符串空白和特殊字符后：".$e;           //输出最后结果
?>
```

（2）将该文件存储于\MR\02\083\文件夹下，命名为 index.php。运行结果如图 2.57 所示。

秘笈心法

心法领悟 083：使用 ltrim()函数和 rtrim()函数的相关说明。

在使用这两个函数时，默认情况下是只去除左边或者右边的空格，想要去除特殊字符需要在函数里注明。

实例 084　对论坛帖子的内容进行转义

光盘位置：光盘\MR\02\084

高级　趣味指数：★★★★

实例说明

论坛帖子中的单引号和双引号在未经转义的情况下输出会出现问题，特别是在输出 SQL 语句时。本实例通过 addslashes()函数来实现转义字符串，运行结果如图 2.58 所示。

未经转义的字符串：PHP数据库插入语句：INSERT INTO table(`id`,`name`)VALUES('1','YangMing')
经转义后的字符串：PHP数据库插入语句：INSERT INTO table(`id`,`name`)VALUES(\'1\',\'YangMing\')

图 2.58　转义字符串

关键技术

本实例主要是通过 addslashes()函数对指定的 SQL 语句进行自动转义。addslashes 函数可以转义单引号、双引号、反斜杠、NULL 字符，主要用于 SQL 语句，对部分字符进行转义，其语法如下：

```
string addslashes ( string str);
```

参数说明：

string str：SQL 语句。

设计过程

（1）创建 PHP 脚本文件，首先，定义插入型 SQL 语句。然后，将 SQL 语句通过 addslashes()函数转义并显示转义结果，其代码如下：

```
<?php
	$str = "INSERT INTO db(id,user,pwd,last_date)VALUES('1','yang','******','2010-06-03')";	//声明一个字符串变量
	echo "未经转义的字符串：".$str."<br>";	//未经转义
	echo "经转义后的字符串：".addslashes($str);	//经转义
?>
```

（2）将该文件存储于\MR\02\084\文件夹中，命名为 index.php。运行结果如图 2.58 所示。

秘笈心法

心法领悟 084：addslashes()函数在程序中的应用。

addslashes()函数主要用于数据库 SQL 语句的转义，当数据要插入到数据库之前，有必要应用 addslashes()函数进行字符串转义，以免特殊字符未经转义在插入数据库时出现错误。

实例 085　还原论坛中帖子的内容

光盘位置：光盘\MR\02\085

高级

趣味指数：★★★★

实例说明

既然可以将帖子内容进行转义，当然也可以将帖子内容进行还原。本实例通过 stripslashes()函数来实现还原字符串，运行结果如图 2.59 所示。

```
未经还原的字符串：PHP数据库插入语句：INSERT INTO table(`id`,`name`)VALUES(\'1\',\'YangMing\')
经还原后的字符串：PHP数据库插入语句：INSERT INTO table(`id`,`name`)VALUES('1','YangMing')
```

图 2.59　还原字符串

关键技术

本实例运用 stripslashes()函数，将 addslashes()函数转义的字符串进行还原，其语法如下：

```
string stripslashes ( string str);
```

参数说明：

string str：SQL 语句。

设计过程

（1）创建 PHP 脚本文件，首先，定义数据库插入语句的字符串变量。然后，将字符串变量通过 stripslashes()函数还原并显示运行结果，其代码如下：

```
<?php
	$str = "INSERT INTO db(id,user,pwd,last_date)VALUES('1','yang','******','2010-06-03')";	//声明一个字符串变量
	echo "未经还原的字符串：".$str."<br>";	//未经还原字符串输出
	echo "经还原后的字符串：".stripslashes($str)."<br>";	//经还原后的字符串输出
?>
```

（2）将该文件存储于\MR\02\085\文件夹下，命名为 index.php。运行结果如图 2.59 所示。

秘笈心法

心法领悟 085：stripslashes()函数的应用。

stripslashes()函数主要用于数据库 SQL 语句的还原输出。

实例 086　截取论坛标题

光盘位置：光盘\MR\02\086

高级

趣味指数：★★★★

实例说明

本实例通过 substr()函数实现截取论坛标题，运行结果如图 2.60 所示。

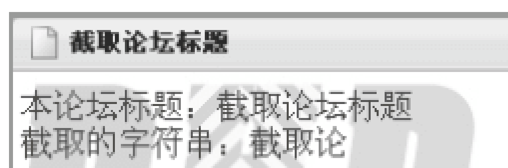


图 2.60　截取论坛标题

关键技术

substr()函数用于从字符串中按照指定位置截取一定长度的字符。通过该函数可以获取某个固定格式字符串中的一部分，如果使用一个正数作为子串起点来调用这个函数，将得到从起点到字符串结束的整个字符串；如果使用一个负数作为子串起点来调用，将得到原字符串尾部的一个子串，字符个数等于给定负数的绝对值，其语法如下：

```
string substr ( string str, int start [, int length])
```

参数 str 用来指定字符串对象；参数 start 用来指定开始截取字符串的位置，如果为负数，则从字符串的末尾开始截取；参数 length 为可选项，指定截取字符的个数，如果其为负数，则表示取到倒数第 length 个字符。

设计过程

（1）创建 PHP 脚本文件。首先，定义变量$a 并将网页标题赋给$a。然后，通过 substr()函数对字符串$a 进行截取，并输出截取内容，其代码如下：

```
<?php
	$b = "截取论坛标题";								//设定字符串
	$a = "<title>$b</title>";							//设定网页标题
	echo $a;										//输出字符串
	echo "本论坛标题：".$b."<br>";						//输出论坛标题
	$b = substr($b,0,12);								//截取论坛标题
	echo "截取的字符串：".$b;							//输出截取内容
?>
```

（2）将该文件存储于\MR\02\053\文件夹下，命名为 index.php。运行结果如图 2.60 所示。

秘笈心法

心法领悟 086：substr()函数应用的注意事项。

substr()函数的参数 start 的指定位置是从 0 开始计算的，即字符串中的第 1 个字符表示为 0。本函数在截取中文字符串时，可能会出现乱码，所以建议在对中文字符串进行截取时应用 mb_substr()函数。

实例 087　分割、合成字符串

光盘位置：光盘\MR\02\087

高级

趣味指数：★★★★

实例说明

字符串的分割、合成对于程序员来说是十分重要的，在应用方面也十分广泛。例如，在遍历数据库查找匹

配项时就会用到合成函数，而购物车程序一般会用到分割函数。本实例通过 explode()函数和 implode()函数实现分割、合成字符串，运行结果如图 2.61 所示。

原字符串：你好PHP 很高兴认识你。
经分割后的数组为：Array ([0] => 你好PHP [1] => 很高兴认识你。)
合成后的字符串：你好PHP很高兴认识你。

图 2.61　分割、合成字符串

关键技术

字符串的分割与合成应用的是 explode()函数和 implode()函数。

（1）explode()函数

explode()函数按照指定的规则对一个字符串进行分割，返回值为数组，其语法如下：

```
array explode ( string separator, string string [, int limit]);
```

参数 string string 表示分割条件；separator 表示指定的分隔符；int limit 表示指定生成数组的最大值。

（2）implode()函数

implode()函数与 explode()函数配合使用，将数组的内容组合成一个字符串，其语法如下：

```
string implode ( string glue, array pieces);
```

参数 glue 是字符串类型，指定分隔符；参数 pieces 是数组类型，指定要被合并的数组。

设计过程

（1）创建 PHP 脚本文件。首先，定义字符串变量，将字符串内容用指定的分隔符分隔。然后，利用 explode()函数，将字符串变量按照规定的分隔符分割，并保存在数组变量$explode 中。最后，分别利用 prinr_r()函数和 implode()函数输出数组和字符串，其代码如下：

```
<?php
    $str = "你好 PHP  很高兴认识你。";                    //定义字符串
    $explode = explode(" ",$str);                       //分割字符串
    echo "原字符串：".$str."<br>";                       //输出原字符串
    echo "经分割后的数组为：";
    print_r($explode);                                  //打印分割后的数组
    echo "<br>合成后的字符串：";
    echo implode($explode);                             //将数组合成为字符串
?>
```

（2）将该文件存储于\MR\02\087\文件夹下，命名为 index.php。运行结果如图 2.61 所示。

秘笈心法

心法领悟 087：explode()函数和 implode()函数哪个应用更广泛？

二者相比，implode()函数应用相对广泛，所有的数组变量，都可以用它进行合成。

实例 088　查询关键字描红

光盘位置：光盘\MR\02\088

高级

趣味指数：★★★★

实例说明

关键字描红技术广泛应用于站内搜索或者高级搜索中，是一项很常用的技术。本实例通过函数 str_replace()实现查询关键字描红，运行结果如图 2.62 所示。

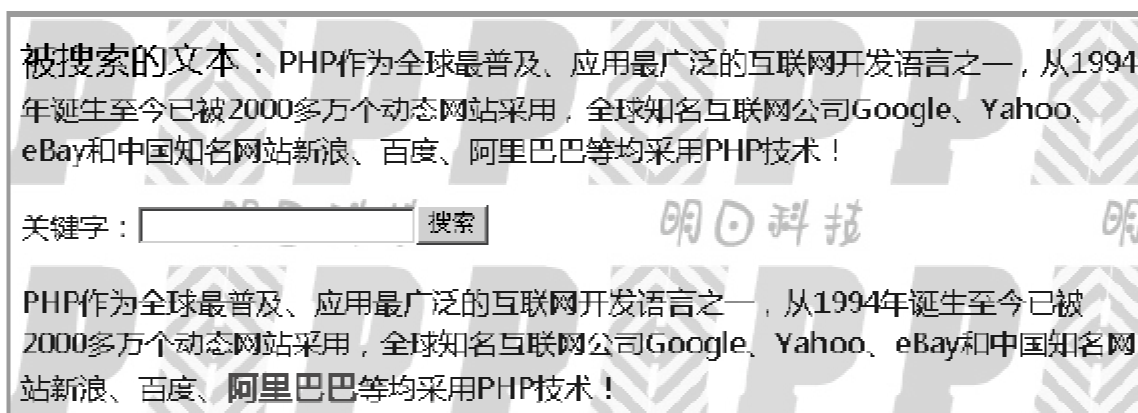


图 2.62　查询关键字描红

关键技术

str_replace ()函数可用于执行字符串替换操作，其语法如下：

```
mixed str_replace ( mixed search, mixed replace, mixed subject [, int &count] )
```

参数说明：

search：必要参数，指定需要查找的字符串。

replace：必要参数，指定替换的值。

subject：必要参数，指定查找范围。

count：可选参数，获取执行替换的数量。

设计过程

（1）创建 PHP 脚本文件，当单击“搜索”按钮时，将通过 POST 方式接收到数据，首先对数据进行描红处理，然后，应用 str_replace()函数对指定字符串信息的指定内容进行替换，其代码如下：

```
<?php
if($_POST[sub]){
    $a = "<b style='color:red;font-size:18px;'>".$_POST[text]."</b>";          //对提交的关键字进行描红
    echo str_replace("$_POST[text]",$a,$str);                                  //完成字符串的替换操作
}
?>
```

（2）将该文件存储于\MR\02\088\文件夹下，命名为 index.php。运行结果如图 2.62 所示。

秘笈心法

心法领悟 088：更多实现关键字描红的函数。

不仅可以通过 str_replace()函数实现关键字的描红操作，还可以通过 str_ireplace()函数实现对关键字的描红功能，并且该函数不区分大小写。

实例 089	统计查询关键字的出现次数 光盘位置：光盘\MR\02\089	高级 趣味指数：★★★★

实例说明

在站内搜索中，往往要求列出符合条件的关键字有多少个。本实例通过字符串函数 substr_count()统计查询关键字的出现次数，运行结果如图 2.63 所示。

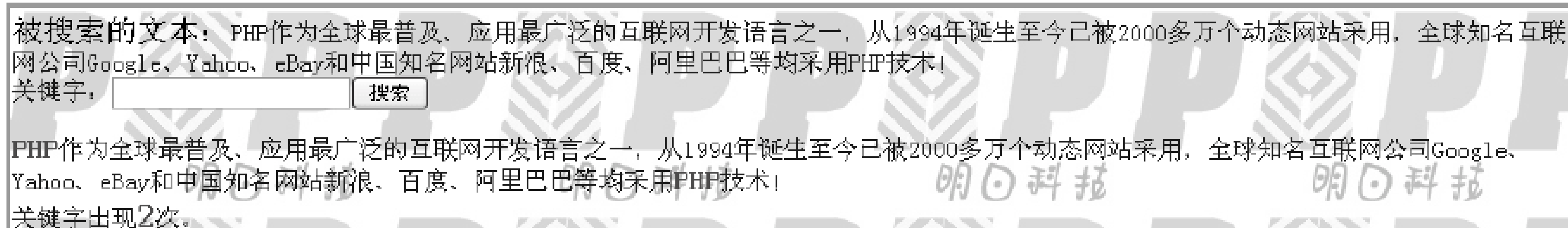


图 2.63　统计查询关键字的出现次数

关键技术

本实例通过 substr_count()函数检索字符串子串的出现次数。substr_count()函数的语法如下：

```
int substr_count ( string haystack, string needle )
```

参数说明：

string haystack：源字符串。

string needle：字符串子串。

设计过程

（1）创建 PHP 脚本文件，首先，定义字符串变量，编写<form>表单。当单击“搜索”按钮时，利用 substr_count()函数输出指定字符串子串的出现次数，其代码如下：

```
<?php
        echo "<a style='font-size:20px; color:#000000'>被搜索的文本：</a>";            //定义字符串输出
        $str = "PHP 作为全球最普及、应用最广泛的互联网开发语言之一，从 1994 年诞生至今已被 2000 多万个动态网站采用，全球知名互联网公司 Google、Yahoo、eBay 和中国知名网站新浪、百度、阿里巴巴等均采用 PHP 技术！ ";
        echo $str."<br>";
?>
<form action ="" method="post">                                              //表单
    关键字：<input type="text" name="text"><input type="submit" name="sub" value ="搜索">
</form>
</body>
</html>
<?php
    if($_POST[sub]){                                                         //通过 POST 方式传递参数
        $a = "<b style='color:red;font-size:18px;'>".$_POST[text]."</b>";    //替换后的字符串
        echo str_replace("$_POST[text]",$a,$str)."<br>";                     //替换字符串
        //统计关键字出现次数
        echo "关键字出现<b style='color:red;font-size:22px;'>".substr_count($str,$_POST[text])."</b>次。";
    }
?>
```

（2）将该文件存储于\MR\02\089\文件夹下，命名为 index.php。运行结果如图 2.63 所示。

秘笈心法

心法领悟 089：substr_count()函数的实现原理。

substr_count()函数是一种类似全文检索的函数，其原理是检索查找匹配的字符串，返回匹配次数。

实例 090	获取上传文件的后缀 光盘位置：光盘\MR\02\090	高级 趣味指数：★★★★

实例说明

文件上传功能几乎是所有网站所必备的，然而文件上传对于服务器来说是具有很大风险的，因此应该对文件大小、文件类型进行限制。本实例通过字符串函数 strrev()获取上传文件的后缀，运行结果如图 2.64 所示。



图 2.64　获取上传文件的后缀

关键技术

PHP 中字符串的逆序输出可以通过循环实现，但最简单的方式是通过函数 strrev()实现。strrev()函数可将字

符串反转，其语法如下：

```
string strrev ( string string );
```

参数说明：

string string：需要反转的字符串。

在本实例中，通过 strrev()函数对上传文件的名称进行反转输出，并通过 explode()函数以“.”为分隔符，对文件名称进行分割，然后再次应用 strrev()函数对数组中的第一个元素值进行返回输出，获取到的就是上传文件的后缀。

设计过程

（1）创建 PHP 脚本文件，当单击“上传”按钮时，首先，利用 POST 方法接收文本框中的信息。然后，利用反转函数将文本框信息数据进行反转，并利用字符串拆分函数以“.”进行分割，然后将分割结果保存在数组中。最后，将指定数据再次反转取得并输出上传文件的后缀名称，其代码如下：

```
<?php
	if($_POST[sub]){                          //通过 post 方式传递参数
		$a = strrev($_POST[text]);            //反转字符串
		$b = explode(".",$a);                 //以点号分割
		$c = strrev($b[0]);                   //反转字符串
		echo "文件后缀为:.$c";                //输出文件后缀
	}
?>
```

（2）将该文件存储于\MR\02\090\文件夹下，命名为 index.php。运行结果如图 2.64 所示。

秘笈心法

心法领悟 090：知识扩展。

利用 strrev()函数检索字符串相对较麻烦，可以使用正则表达式函数 preg_match()来完成该功能。

实例 091	统一上传文件名称的大小写 光盘位置：光盘\MR\02\091	高级 趣味指数：★★★★

实例说明

统一上传文件名称的大小写是十分必要的。例如上传图片文件 img.jpg 和 Img.jpg，如果文件名称未经大小写统一，则上传时可能不会出现覆盖提示。本实例通过字符串函数 strtoupper()和 strtolower()实现统一上传文件名称的大小写，运行结果如图 2.65 所示。

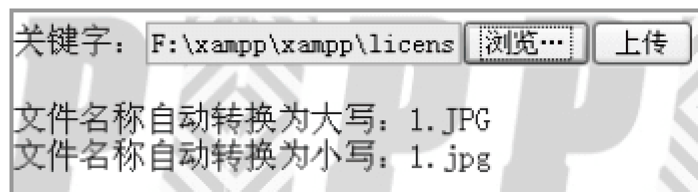


图 2.65　统一上传文件名称的大小写

关键技术

实现字符串的大小写转换，主要通过 strtolower()和 strtoupper()函数实现。

（1）strtolower()函数

使用该函数转换为小写。

```
string strtolower ( string str )
```

（2）strtoupper()函数

使用该函数转换为大写。

```
string strtoupper ( string string )
```

参数 string 为需要转换的字符串。

设计过程

（1）创建 PHP 脚本文件，当单击“上传”按钮时，分别利用转换小写和大写的函数 strtolower()和 strtoupper()将接收到的文本框的信息进行对应的大小写转换，并输出转换结果，其代码如下：

```
<?php
	if($_POST[sub]){                                          //通过 POST 方式传参
		$a = strtoupper($_POST[text]);                    //转换为大写
		echo "文件名称自动转换为大写：".$a."<br>";          //输出
		$b = strtolower($_POST[text]);                    //转换为小写
		echo "文件名称自动转换为小写：".$b;                 //输出
	}
?>
```

（2）将该文件存储于\MR\02\091\文件夹下，命名为 index.php。运行结果如图 2.65 所示。

秘笈心法

心法领悟 091：strtolower()和 strtoupper()函数的应用范围。

在电子商务类网站的后台商品添加页面中，为了使商品的型号以统一的规格显示，经常会使用大小写转换函数，所以掌握这两个函数是很有必要的。

实例 092　论坛中直接输出 HTML 脚本

光盘位置：光盘\MR\02\092

高级

趣味指数：★★★★

实例说明

在论坛文本框里使用 HTML 标签是存在作用效果的，但是对于论坛管理员来说是不希望看到的，因为它会给服务器造成压力。本实例通过 htmlentities()函数实现字符串与 HTML 的转换，运行结果如图 2.66 所示。

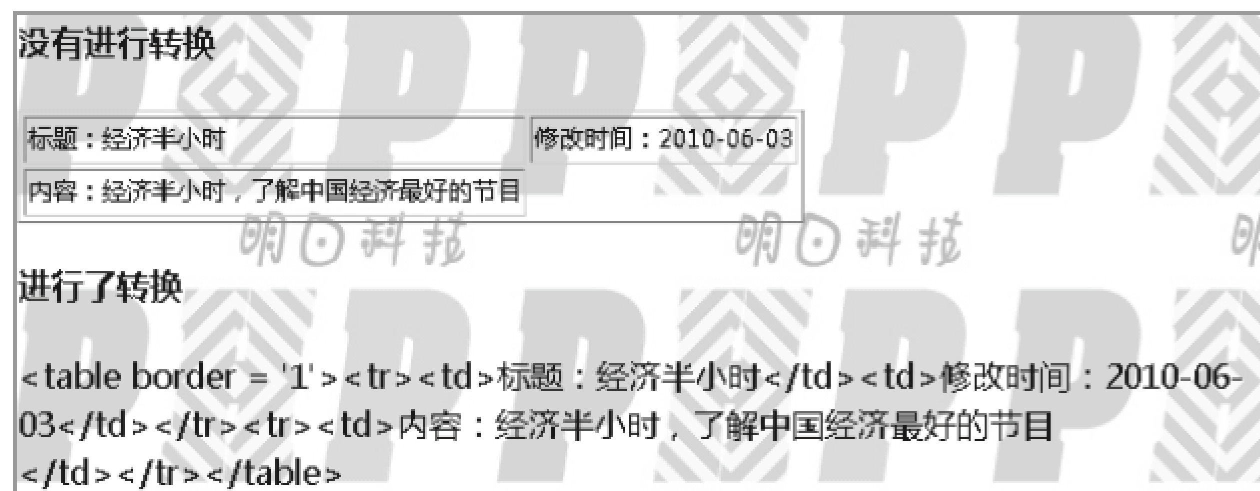


图 2.66　字符串与 HTML 转换

关键技术

PHP 中使 HTML 代码格式化输出可以使用函数 htmllentities()实现。htmllentities()函数的语法如下：

```
string htmlentities(string string,[int quote_style],[string charset]);
```

参数说明：

string string ：需要格式化输出的 html 代码。

int quote_style：可选参数，一般设置为 ENT_QUOTES。

string charset：设置编码格式。

设计过程

（1）创建 PHP 脚本文件，首先，定义字符串变量，变量内容为一段 HTML 标签（创建表格的代码）。然后，利用 htmllentities()函数将字符串变量进行转换并输出转换结果，其代码如下：

```
<?php
	echo "<h2>没有进行转换</h2>";
$str = "<table border = '1'><tr><td>标题：经济半小时</td><td>修改时间：2010-06-03</td></tr><tr><td>内容：经济半小时，了解中国经济最好的节目</td></tr></table>";		//定义 HTML 格式
	echo $str;							//输出字符串
	echo "<h2>进行了转换</h2>";
	echo htmlentities($str,ENT_QUOTES,"GB2312")."<br>";		//转换 HTML
?>
```

（2）将该文件存储于\MR\02\092\文件夹下，命名为 index.php。运行结果如图 2.66 所示。

秘笈心法

心法领悟 092：使用格式化函数输出的优点。

格式化输出 HTML 可以直接将提交的源码输出并确保源码不被解析，从而可以确保直接在页面中输出 HTML 标签，而不被解析。

实例 093	统计帖子标题的长度 光盘位置：光盘\MR\02\093	高级 趣味指数：★★★★

实例说明

统计字符串长度一般是为了给其他函数的应用做好铺垫。本实例通过 strlen()函数获取帖子标题的长度，运行结果如图 2.67 所示。

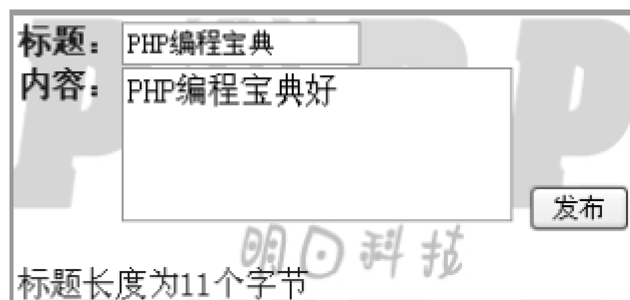


图 2.67　统计帖子标题的长度

关键技术

strlen()函数用于获取字符串的长度，其语法如下：

```
int strlen ( string str);
```

参数说明：

str：指定字符串。

设计过程

（1）创建 PHP 脚本文件，当单击“发布”按钮时，利用 strlen()函数获取通过 POST 方式接收到的数据长度，其代码如下：

```
<?php
	if($_POST[sub]){						//通过 POST 方式传参
		echo "标题长度为".strlen($_POST[text])."个字节";		//获取字符串长度
	}
?>
```

（2）将该文件存储于\MR\02\093\文件夹下，命名为 index.php。运行结果如图 2.67 所示。

秘笈心法

心法领悟 093：使用 strlen()函数的注意事项。

使用 strlen()函数获取字符串长度时，如果字符串中存在空格，则空格也会被计算在内。

实例 094 计算论坛帖子的字节数

光盘位置：光盘\MR\02\094

高级

趣味指数：★★★★

实例说明

使用 strlen()函数可以统计论坛标题的长度，同样可以对论坛中帖子的字节数进行统计。本实例通过 strlen()函数对论坛中帖子的字节数进行统计，运行结果如图 2.68 所示。

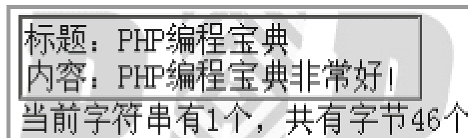


图 2.68 计算论坛帖子的字节数

关键技术

本实例同样应用 strlen()函数统计字符串的长度，并且应用自定义函数对提交的源代码进行处理，保证源代码以原形式输出，而不会输出执行文件。自定义函数的语法如下：

```
function unhtml($content){
      $content=htmlspecialchars($content);
      $content=str_replace(chr(13),"<br>",$content);
      $content=str_replace(chr(32)," ",$content);
    return trim($content);
}
```

设计过程

（1）创建 PHP 脚本文件，首先定义自定义函数，然后定义字符串变量，接着应用 strlen()函数统计帖子内容的字节数，并且通过自定义函数对帖子内容进行转换，其代码如下：

```
<?php
function unhtml($content){
      $content=htmlspecialchars($content);
      $content=str_replace(chr(13),"<br>",$content);
      $content=str_replace(chr(32)," ",$content);
    return trim($content);
 }
$str = "标题：PHP 编程宝典<br>内容：PHP 编程宝典非常好！ ";
?>
<table border="1" bordercolor="#FF0000" cellspacing="0"><tr><td>
<?php echo $str;?>
</td></tr></table>
<?php
echo "共有字节".strlen(unhtml($str))."个";
?>
```

（2）将该文件存储于\MR\02\094\文件夹下，命名为 index.php。运行结果如图 2.68 所示。

秘笈心法

心法领悟 094：自定义函数 unhtml()。

自定义函数 unhtml()主要用于对提交的源代码进行处理，保证源代码以原形式输出，而不会输出执行文件。

实例 095　实现帖子内容的定位查找

光盘位置：光盘\MR\02\095

高级

趣味指数：★★★★

实例说明

用微软 Windows 操作系统查找文件，可以使用 Ctrl+F 快捷键实现快速定位查找。本实例通过 strpos()函数模拟这一功能实现定位查找，运行结果如图 2.69 所示。



图 2.69　实现帖子内容定位查找

关键技术

在 PHP 中实现特殊字符的查找和定位可以通过函数 strpos()来实现，函数语法如下：

```
int strops(string str1 ,string str2, [int offset])
```

此函数的作用是在 str1 中查找 str2，并返回查找到的位置。strpos()函数实现的原理比较整个要查找的字符串，并返回字符串 str2 第 1 次出现的位置。

设计过程

（1）创建 PHP 脚本文件，当单击“定位查找”按钮时，首先定义一个字符串变量$email。然后，将接收到的文本框数据信息进行描红处理。最后利用 strpos()函数实现帖子内容的定位查找，其代码如下：

```
<?php
	if($_POST[sub]){
		$email = '吉林省明日科技有限公司是一家以计算机软件技术为核心的高科技型企业';
		$str = $_POST[te];
		$str1 = "<b style='color:red;'>".$str."</b>";
		echo str_replace($str,$str1,$email)."<br>";
		echo "查找内容从<b style='color:red;'>".strpos($email,$str)."</b>字节开始，字符串长度为<b style='color:red;'>".strlen($str)."</b>个字节";
	}
?>
```

（2）将该文件存储于\MR\02\095\文件夹下，命名为 index.php。运行结果如图 2.69 所示。

秘笈心法

心法领悟 095：知识扩展。

strpos()函数一般与 strlen()函数联合使用，得到的两个结果相当于 substr()函数的两个位置参数。

实例 096　去除帖子标题的首尾空格

光盘位置：光盘\MR\02\096

高级

趣味指数：★★★★

实例说明

用户在编写代码时也许会在不经意间多输入了一个或多个空格，使得程序无论怎么调试都无法正常运行。本实例通过 trim()函数实现去除帖子标题的首尾空格，运行结果如图 2.70 所示。

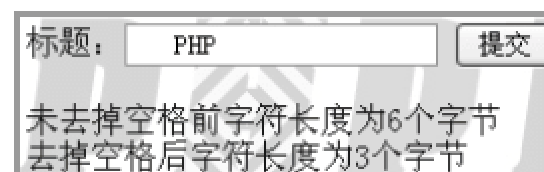


图 2.70　去除帖子标题的首尾空格

关键技术

在 PHP 中去除字符串的首尾空格，可以使用函数 trim()实现。trim()函数的语法如下：

```
string trim(string str);
```

参数说明：

string str：需要过滤空格的字符串。

设计过程

（1）创建 PHP 脚本文件，当单击“提交”按钮时，利用 strlen()和 trim()函数的联合操作，实现输出去除首尾空格前、后的字符串的长度，其代码如下：

```
<?php
	if($_POST[sub]){                                                            //通过 POST 方式传递参数
		echo "未去掉空格前字符长度为".strlen($_POST[text])."个字节<br>";      //输出结果
		echo "去掉空格后字符长度为".strlen(trim($_POST[text]))."个字节";      //输出结果
	}
?>
```

（2）将该文件存储于\MR\02\096\文件夹下，命名为 index.php。运行结果如图 2.70 所示。

秘笈心法

心法领悟 096：知识扩展。

PHP 还提供有单纯去除字符串左边或者右边空格的函数，详细说明参见实例 083。

实例 097　货币数据的格式化输出

光盘位置：光盘\MR\02\097

高级　趣味指数：★★★★

实例说明

货币数据不同于整型数据，其是存在一定格式的。本实例通过函数 number_format()实现金额的格式化输出，运行结果如图 2.71 所示。



图 2.71　金额的格式化输出

关键技术

金额的格式化输出通过函数 number_format()实现，其语法及说明如下：

```
string number_format ( float number , int decimals)
string number_format ( float number, int decimals, string dec_point, string thousands_sep)
```

number_format()函数返回参数 number 格式化后的字符串，该函数可以有 1 个、2 个或是 4 个参数，但不能是 3 个参数。如果只有 1 个参数 number，number 格式化后会舍去小数点后的值，且每一千以逗号（,）隔开；如果有 2 个参数，number 格式化后会到小数点第 decimals 位，且每一千就会以逗号来隔开；如果有 4 个参数，

number 格式化后会到小数点第 decimals 位，dec_point 用来替代小数点（.），thousands_sep 用来替代每一千隔开的逗号（,）。

设计过程

（1）创建 PHP 脚本，当单击“格式化”按钮时，程序自动将文本框中的信息进行格式化，代码如下：

```
<?php
	if($_POST[sub]){					//通过 POST 方式传参
		echo number_format($_POST[text],2);		//格式化数字
	}
?>
```

（2）将该文件存储于\MR\02\097 文件夹下，命名为 index.php。运行结果如图 2.71 所示。

秘笈心法

心法领悟 097：函数应用。

开发电子商务类网站，函数 number_format()出现的频率是百分之百的，所以掌握此函数是十分必要的。

实例 098	日期、时间的格式化输出 光盘位置：光盘\MR\02\098	高级 趣味指数：★★★★

实例说明

在 PHP 程序设计中，经常用到对日期、时间的格式化输出。本实例通过函数 date()实现日期、时间的格式化输出，运行结果如图 2.72 所示。

显示当前日期、时间 2010-06-24 13:31:04

图 2.72　日期、时间的格式化输出

关键技术

本实例的关键点是 date()函数的灵活运用，详细说明详见实例 035。

设计过程

（1）创建 PHP 脚本，当单击“显示当前日期、时间”按钮时，利用 date()函数根据传入的指定的参数格式实现格式化输出，代码如下：

```
<?php
	date_default_timezone_set("Asia/ShangHai");		//设置时区
	If($_GET[date] == 1){					//通过 GET 方式取得参数
		echo date("Y-m-d H:i:s");			//格式化输出时间和日期
	}
?>
```

（2）将该文件存储于\MR\02\098 文件夹下，命名为 index.php。运行结果如图 2.72 所示。

秘笈心法

心法领悟 098：获取当地时间。

获取当地时间时一定要首先设置时区，否则 PHP 默认使用的是英国伦敦的零时区，而在我们本地使用的是北京时间，即东八区的时间。设置时区时，可使用 date_default_timezone_set("Asia/ShangHai")语句。

实例 099 比对论坛中帖子的相似度

光盘位置：光盘\MR\02\099

高级

趣味指数：★★★★

实例说明

在论坛程序中比较两个帖子是否相同可以屏蔽重复帖子，节省资源空间。本实例通过字符串函数 similar_text()比较两个帖子的相似度，运行结果如图 2.73 所示。

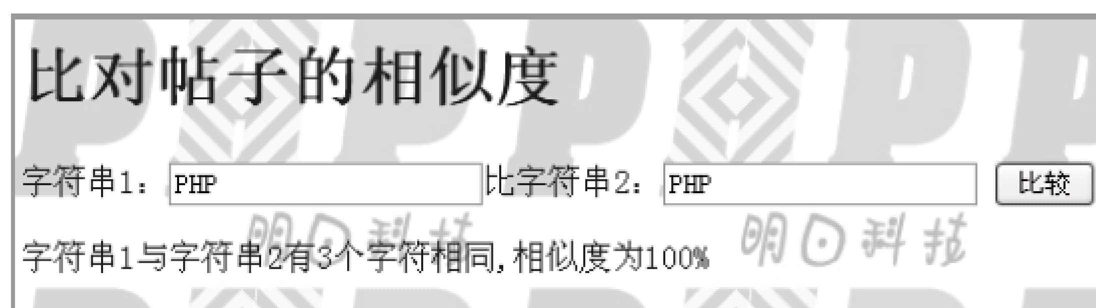


图 2.73　比对论坛中帖子的相似度

关键技术

PHP 中字符串相似度的获取可以通过函数 similar_text()来实现。其语法格式如下：

```
int similar_text(sting str1,string str2,[double precent])
```

该函数用于比较字符串 str1 和字符串 str2 的相似程度，函数的返回结果是字符串 str1 和字符串 str2 的相同字符的个数，而可省参数 precent 的值是这两个参数的相似度。

设计过程

（1）创建 PHP 脚本，当单击“比较”按钮时，首先，分别获取字符串 1 和字符串 2 中的数据，然后，利用字符串函数 simlar_text()实现字符串相似度的获取，代码如下：

```
<?php
	if($_POST[sub]){                                                //通过 POST 方式接收参数
		$a = similar_text($_POST[te],$_POST[tex]);                  //判断两个字符串的相似度
		if(strlen($_POST[te]) >= strlen($_POST[tex])){              //计算百分比
			echo "字符串 1 与字符串 2 有".$a."个字符相同,相似度为".($a/strlen($_POST[te])*100)."%";
		}else{
			echo "字符串 1 与字符串 2 有".$a."个字符相同,相似度为".($a/strlen($_POST[tex])*100)."%";
		}
	}
?>
```

（2）将该文件存储于\MR\02\099 文件夹下，命名为 index.php。运行结果如图 2.73 所示。

秘笈心法

心法领悟 099：similar_text()函数的原理。

similar_text()函数的原理是对应比较每个字节并根据结果计算百分比。

实例 100 对用户注册的密码进行加密和解密

光盘位置：光盘\MR\02\100

高级

趣味指数：★★★★

实例说明

对于电子商务网站的用户来说，最重要的就是账号密码的安全性。如果账号密码未经过加密处理，则会大

大增加安全隐患，给非法用户以可乘之机。本实例是通过字符串函数 base64_encode()和 base64_decode()实现对密码的加密和解密，运行结果如图 2.74 所示。

图 2.74　对用户注册的密码进行加密和解密

关键技术

对字符串的 base64 编码可以通过 PHP 的预定义函数 base64_encode()实现，其语法如下：

```
string base64_encode(string str)
```

该函数实现对字符串 str 的 base64 编码。

对已经进行 base64 编码的字符串进行解码用函数 base64_decode()实现，其语法如下：

```
string base64_decode(string str)
```

该函数实现对已经进行 base64 编码的字符串进行解码。

设计过程

（1）创建 PHP 脚本，当单击“提交”按钮时，通过字符串函数 base64_encode()和 base64_decode()实现对字符串的加密和解密，代码如下：

```
<?php
	if($_POST[sub]){                                    //通过 POST 方式接收参数
		echo "经加密<hr>";
		echo "用户名：".$_POST[text];
		$str = base64_encode($_POST[pwd]);          //对密码进行加密
		echo "密码：". $str;
		echo "<hr>";
		echo "经解密";
		echo "<hr>";
		echo "用户名：".$_POST[text];
		echo "密码：". base64_decode($str);          //对密码进行解密
	}
?>
```

（2）将该文件存储于\MR\02\100 文件夹下，命名为 index.php。运行结果如图 2.74 所示。

秘笈心法

心法领悟 100：加密函数的应用。

base64_encode()加密技术在邮件系统建设等方面也得到了广泛的应用。

实例 101	保护 URL 地址中传递的参数 光盘位置：光盘\MR\02\101	高级 趣味指数：★★★★

实例说明

表单的提交方式有两种：GET 方式和 POST 方式。GET 方式属于明文地址栏传参，存在安全隐患。本实例

通过 urlencode()函数实现保护 URL 地址中传递的参数，运行结果如图 2.75 所示。



图 2.75　保护 URL 地址中传递的参数

关键技术

PHP 中字符串的 url 编码通过函数 urlencode()实现，其语法如下：

```
string urlencode(string str)
```

该函数实现对字符串 str 的 url 编码。

设计过程

（1）创建 PHP 脚本，定义字符串变量，将此字符串变量通过函数 urlencode()进行编码，然后将编码后的字符串变量以拼接地址栏参数的形式定义在超链接中，代码如下：

```
<?php
	$str = "id = 1";                                  //定义字符串
	$str = urlencode($str);                           //对字符串进行编码
?>
<a href="index.php?<?php echo $str;?>">对地址栏参数加密</a>     //链接
```

（2）将该文件存储于\MR\02\101 文件夹下，命名为 index.php。运行结果如图 2.75 所示。

秘笈心法

心法领悟 101：知识拓展。

经编码的 URL 参数是可以通过 urldecode()函数进行解码的，详细说明详见本实例。即使不对 URL 编码后的参数进行解码，同样也可以获取到传递的参数值。

实例 102　解析 URL 地址中传递的编码参数

光盘位置：光盘\MR\02\102

高级　趣味指数：★★★★

实例说明

本实例通过 urldecode()函数解析 URL 地址中传递的参数值，运行结果如图 2.76 所示。

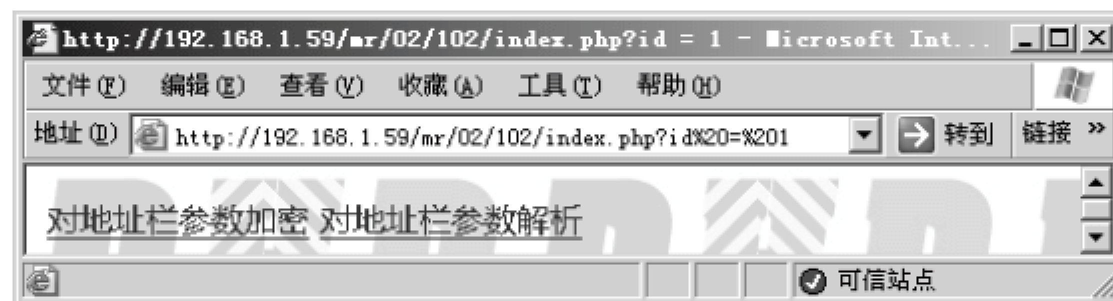


图 2.76　解析 URL 地址中传递的编码参数

关键技术

对已经进行 URL 编码的字符串进行解码使用 urldecode()函数，语法如下：

```
string urldecode ( string str);
```

设计过程

（1）创建 PHP 脚本。首先，定义字符串变量并为此字符串变量赋值。然后，将字符串变量进行 URL 编码处理。最后，单击超链接，并将连接地址用 urldecode()函数解码，代码如下：

```
<?php
	$str = "id = 1";                                    //定义字符串
	$str = urlencode($str);                             //对 url 进行编码
	$str1 = urldecode($str);                            //对 url 进行解析
?>
<a href="index.php?<?php echo $str1;?>">对地址栏参数解析</a>          //地址栏传参
```

（2）将该文件存储于\MR\02\102 文件夹下，命名为 index.php。运行结果如图 2.76 所示。

秘笈心法

心法领悟 102：URL 编码是否可以应用于防止 SQL 注入。

URL 编码不能有效地防止 SQL 注入，原因在于 URL 编码可以解析。

实例 103　转义帖子中的特殊字符

光盘位置：光盘\MR\02\103

高级　趣味指数：★★★★

实例说明

实例 043 已经向用户介绍过“\”转义特殊字符的方法。下面通过函数 quotemeta()实现转义特殊字符，运行结果如图 2.77 所示。

转换前：$.\+*?[]^()mingri
转换后：\$\.\\\+*\?\[\]\^\(\)mingri

图 2.77　特殊字符的转换

关键技术

为了方便特殊字符的原样输出，PHP 提供了函数 quotemeta()，其语法如下：

```
string quotemeta(string str)
```

此函数的作用是在字符串 str 中的某些字符前面加上反斜杠“\”，这些特殊字符主要有“.”、“\”、“+”、“*”、“?”“[”、“]”、“^”、“(”、“)”、“$”，其中如果“$”后面的字符与“$”共同构成了一个合法的变量名，则“$”前不加“\”。

设计过程

（1）创建 PHP 脚本。首先，定义字符串变量并将一系列特殊符号赋给变量。然后，通过函数 quotemeta()将字符串变量进行转义输出，代码如下：

```
<?php
	$s="mingri";                                        //定义字符串变量
	$str="$.\+*?[]^()$s";                               //定义字符串变量
		echo"转换前：".$str."<br>";
		echo "转换后：".quotemeta($str);                  //转换特殊字符
?>
```

（2）将该文件存储于\MR\02\103 文件夹下，命名为 index.php。运行结果如图 2.77 所示。

秘笈心法

心法领悟 103：解决无法原样输出字符的问题。

在进行 PHP 项目开发过程中，经常会遇到一些无法原样输出的字符，解决办法就是通过转义字符进行转义。

实例 104 获取任意字符的 ASCII 码

光盘位置：光盘\MR\02\104

高级

趣味指数：★★★★

实例说明

敲击计算机键盘上的任意一个按键都要经过 ASCII 码转换后显示到屏幕上。本实例通过字符串函数 ord()实现将字符转换为 ASCII 码，运行结果如图 2.78 所示。



图 2.78 获取任意字符的 ASCII 码

关键技术

PHP 中获取字符的 ASCII 码应使用 ord()函数，其语法如下：

```
int ord ( string string);
```

该函数预期返回一个整型数值。

设计过程

（1）创建 PHP 脚本，当单击“转换”按钮时，通过字符串函数 ord()获取指定字符串的 ASCII 码，代码如下：

```
<?php
	if($_POST[sub]){                                    //通过 POST 方式传递参数
		echo "$_POST[text]的 ASCII 码为：";              //输出
		echo ord($_POST[text]);                         //ASCII 转码
	}
?>
```

（2）将该文件存储于\MR\02\104 文件夹下，命名为 index.php。运行结果如图 2.78 所示。

秘笈心法

心法领悟 104：知识拓展。

像 HTML 标签“
”等都可以在 ASCII 码中找到相同功能的 ASCII 码值。

实例 105 查找指定 ASCII 码对应的字符

光盘位置：光盘\MR\02\105

高级

趣味指数：★★★★

实例说明

本实例是通过字符串函数 chr()实现将 ASCII 码转换为字符，运行结果如图 2.79 所示。



图 2.79 查找指定 ASCII 码对应的字符

关键技术

将 ASCII 码转换为字符可以通过函数 chr()实现，其语法如下：

```
string chr ( int ascii);
```

该函数可将 ASCII 序数值转换成对应的字符，与 ord()函数成对照。

设计过程

（1）创建 PHP 脚本。当单击“转换”按钮时，首先，利用 POST 方法获取文本框中传入的数据信息。然后，将数据信息利用 chr()函数转换为 ASCII 码并实现输出，代码如下：

```
<?php
    if($_POST[sub]){                                  //通过 POST 方式传参
        echo "$_POST[text]的 ASCII 码为：";           //输出
        echo chr($_POST[text]);                       //ASCII 对应的字符
    }
?>
```

（2）将该文件存储于\MR\02\105 文件夹下，命名为 index.php。运行结果如图 2.79 所示。

秘笈心法

心法领悟 105：ASCII 码的优点。

ASCII 码在程序开发中的使用相当广泛，在某些时候将字符转换成 ASCII 码而被系统识别，既能保证安全性，又能获取更好的支持。

实例 106	合理定义上传文件的名称 光盘位置：光盘\MR\02\106	高级 趣味指数：★★★★

实例说明

为了使上传文件的名称不至于与服务器已有文件的名称重名，合理定义上传文件的名称就显得十分重要。本实例通过 rand()函数合理定义上传文件的名称，运行结果如图 2.80 所示。



图 2.80　合理定义上传文件的名称

关键技术

在 PHP 中应用 move_uploaded_file()函数实现文件的上传。但是，在执行文件上传之前，为了防止潜在的攻击对原本不能通过脚本交互的文件进行非法管理，可以先应用 is_uploaded_file()函数判断指定的文件是否是通过 HTTP POST 上传的，如果是，则返回 TRUE。

（1）is_uploaded_file()函数

该函数用于判断指定的文件是否是通过 HTTP POST 上传的，其语法如下：

```
bool is_uploaded_file ( string filename )
```

参数 filename 必须指定类似于$_FILES['filename']['tmp_name']的变量，不可以使用从客户端上传的文件名$_FILES['filename']['name']。

通过 is_uploaded_file()函数对上传文件进行判断，可以确保恶意的用户无法欺骗脚本去访问本不能访问的文件，例如/etc/passwd。

（2）move_uploaded_file()函数

使用该函数将文件上传到服务器中指定的位置。如果成功返回 TRUE，否则返回 FALSE，语法如下：

```
bool move_uploaded_file ( string filename, string destination )
```

参数 filename 指定上传文件的临时文件名，即$_FILES[tmp_name]；参数 destination 指文件上传后保存的新路径和名称。如果参数 filename 不是合法的上传文件，不会出现任何操作，move_uploaded_file()将返回 FALSE；如果参数 filename 是合法的上传文件，但由于某些原因无法移动，也不会出现任何操作，move_uploaded_file()将返回 FALSE，此外还会发出一条警告。

（3）rand()函数

使用该函数生成的随机整数作为上传文件的新名称。

设计过程

创建 index.php 文件，添加表单，设置文件域、提交按钮，使用 POST 方法，设置 enctype="multipart/form-data"，通过$_FILES 获取上传文件的相关信息，通过 move_upload_file()函数完成图片上传，其代码如下：

```
<?php
if(!empty($_FILES[up_picture][name])){                                      //判断上传内容是否为空
 if($_FILES['up_picture']['error']>0){                                      //判断文件是否可以上传到服务器
        echo "上传错误:";
        switch($_FILES['up_picture']['error']){
                case 1:
                        echo "上传文件大小超出配置文件规定值";
                break;
                case 2:
                        echo "上传文件大小超出表单中约定值";
                break;
                case 3:
                        echo "上传文件不全";
                break;
                case 4:
                        echo "没有上传文件";
                break;
        }
 }else{
        if(!is_dir("./upfile/")){                                          //判断指定目录是否存在
                mkdir("./upfile/");                                        //创建目录
        }
        $path='./upfile/'.rand().$_FILES['up_picture']['name'];            //定义文件名称和存储位置
        if(is_uploaded_file($_FILES['up_picture']['tmp_name'])){           //是否是 HTTP POST 上传
                if(!move_uploaded_file($_FILES['up_picture']['tmp_name'],$path)){        //执行上传
                        echo "上传失败";
                }else{
                        echo "文件".time().$_FILES['up_picture']['name']."上传成功，大小为：".$_FILES['up_picture']['size'];
                }
        }else{
                echo "上传文件".$_FILES['up_pictute']['name']."不合法！";
        }
 }
}
?>
```

秘笈心法

心法领悟 106：上传文件的命名问题。

上传文件如果和已有文件重名，文件可能会相互替换，给网站造成损失。所以确保上传文件名称的唯一性是非常必要的，也可以通过 time()时间戳来定义上传文件的名称。

实例 107　通过异或方式对用户注册密码进行加密　高级

光盘位置：光盘\MR\02\107　趣味指数：★★★★

实例说明

本实例通过异或方式对用户注册的密码进行加密，并且输出加密后的密码，运行结果如图 2.81 所示。

图 2.81　通过异或方式对用户注册密码进行加密

关键技术

通过异或方式对字符串加密的原理：当一个字符串 A 与另一个字符串 B 进行异或运算后会产生一个字符串 C。

设计过程

（1）创建 PHP 脚本。首先，定义字符串变量并为此字符串变量赋值。然后，通过异或方式对字符串变量进行加密并输出加密结果，代码如下：

```
<?php
	if($_POST[sub]){                                              //通过 POST 方式传参
		$str = "0123456798";                                      //定义字符串
		echo "用户名：".$_POST[text]."<br>";
		echo "密  码：".($_POST[pwd]^$str);             //异或加密
	}
?>
```

（2）将该文件存储于\MR\02\107 文件夹中，命名为 index.php。运行结果如图 2.81 所示。

秘笈心法

心法领悟 107：比较异或方式与 md5()加密技术。

PHP 中对字符串加密的方式有很多种，像 md5()加密等。异或方式与 md5()加密方式的本质区别是 md5()加密后不能解密，异或方式可以。相对而言，md5()加密更加安全。

实例 108　统一英文注册用户首字母的大小写　高级

光盘位置：光盘\MR\02\108　趣味指数：★★★★

实例说明

网站的国际化已经不是一个炙手可热的话题，但是由于地区差异，在使用英文填写注册信息时，首字母的

大小写往往得不到统一。本实例利用函数 ucfirst()实现将英文注册用户首字母统一为大写，运行结果如图 2.82 所示。



图 2.82　统一英文注册用户首字母的大小写

关键技术

在 PHP 开发中，定义首字母大写的方法有很多，但是所有的方法都没有使用 PHP 预定义函数 ucfirst()简单。ucfirst()的语法如下：

```
string ucfirst ( string str );
```

参数说明：

string str：定义字符串。

设计过程

（1）创建 PHP 脚本。当单击“提交”按钮时，首先，利用 POST()方法获取从文本框传递进来的数据。然后，将获取的数据经 ucfirst()函数转换首字母大写，最后，输出经转换后的数据，代码如下：

```
<?php
    if($_POST[sub]){                              //通过 POST 方式传参
        $a = $_POST[text];                        //接收参数
        echo "首字母统一为大写:";
        echo ucfirst($a);                         //转换单词首字母
    }
?>
```

（2）将该文件存储于\MR\02\108 文件夹中，命名为 index.php。运行结果如图 2.82 所示。

秘笈心法

心法领悟 108：知识拓展。

实现字符串首字母大小写转换还可以通过转码来完成。首先利用函数 substr()提取出字符串的首字母，之后利用函数 ord()获取该字母的 ASCII 码，再根据大小写字母之间相差 32 来实现大小写字母之间的 ASCII 码转换，最后利用函数 chr()将转变后的 ASCII 码转变为字母即可。

实例 109	字符串与数字之间的转换 光盘位置：光盘\MR\02\109	高级 趣味指数：★★★★

实例说明

来看两个变量$a = 123;和$a = ‘123’;。前者是整型变量，后者是字符型变量。本实例利用函数使字符型变量转换为整型变量，整型变量转换为字符型变量，运行结果如图 2.83 所示。

转换前类型为：string，原字符串输出：123
转换后输出：123，转换后类型为：integer
转换前类型为：integer，原字符串输出：123
转换后输出：123，转换后类型为：string

图 2.83　字符串与数字之间的转换

关键技术

在 PHP 中，将数字转换为字符串可以通过函数 strval()实现，其语法如下：

```
string strval ( mixed var)
```

该函数用于返回变量 var 对应的 string 值。

设计过程

（1）首先创建脚本文件，定义字符串变量并将数值型字符串赋给变量。然后通过 strval()函数转换数据，代码如下：

```
<?php
	$str = "123";                                      //定义字符串
	echo "转换前类型为：".gettype($str);               //取得当前数据类型
	echo "，原字符串输出：".$str;
	echo "<br>";
	echo "转换后输出：".intval($str);                  //将字符串转换为整型
	echo "，转换后类型为：".gettype(intval($str));     //取得转换后的数据类型
	echo "<hr>";
	$int = 123;                                        //定义整型变量
	echo "转换前类型为：".gettype($int);               //取得当前类型
	echo "，原字符串输出：".$int;
	echo "<br>";
	echo "转换后输出：".strval($int);                  //转换为字符串
	echo "，转换后类型为：".gettype(strval($int));     //输出转换后的数据类型
?>
```

（2）将该文件存储于\MR\02\109 文件夹下，命名为 index.php。运行结果如图 2.83 所示。

秘笈心法

心法领悟 109：简述 strval()和 intval()函数。

strval()和 intval()函数在实现功能上与强制类型转换机制相同。

实例 110	通过正则表达式对字符串进行匹配查找 光盘位置：光盘\MR\02\110	高级 趣味指数：★★★★

实例说明

字符串匹配查找的方法有很多，下面来了解如何用正则表达式对字符串进行匹配查找。本实例是通过数组函数和正则表达式函数 preg_match()来实现字符串的匹配查找的，运行结果如图 2.84 所示。



图 2.84　利用正则表达式实现字符串的匹配查找

关键技术

灵活运用正则表达式函数是本实例的关键。preg_match()函数用于匹配指定字符串，一般情况下，会与 if 条件语句协同使用，其语法如下：

```
int preg_match ( string pattern, string subject [, array matches] )
```

函数功能：在字符串 subject 中匹配表达式 pattern。函数返回匹配的次数。如果有数组 matches，那么每次

匹配的结果将被存储到数组 matches 中。该函数在匹配成功后就停止继续查找，其返回值是 0 或 1。

设计过程

（1）创建 PHP 脚本。首先，定义数组变量$a 并为$a 赋值。然后，利用 foreach()语句遍历数组以便取得 key 值和 value 值。最后，利用正则表达式函数 preg_match()将通过 POST 方法传递进来的数据进行匹配，匹配成功则输出对应数组的 value 值，代码如下：

```
<?php
	$a = array('mrsoft','明日 mr 科技','www.mrsoft.com');			//定义数组
	foreach($a as $key => $value){						//foreach()循环遍历数组
		$a = $_POST[te];
		if(preg_match("/$a/",$value)){					//正则表达式匹配
			echo $value;
			echo "<br>";
		}
	}
?>
```

（2）将该文件存储于\MR\02\110 文件夹下，命名为 index.php。运行结果如图 2.84 所示。

秘笈心法

心法领悟 110：SQL 语句的匹配问题。

PHP 项目开发过程中可以利用 SQL 语句中的关键字 like 对数据库中的内容进行匹配查找，其实现的是数据库中数据的模糊查询。

实例 111	通过 IP 地址查找主机所在地 光盘位置：光盘\MR\02\111	高级 趣味指数：★★★★

实例说明

IP 地址是互联网身份的象征，通过 IP 地址可以得到用户的相关信息。本实例通过正则表达式验证 IP 地址是否合法，然后实现 IP 地址的格式转换，进而获取 IP 地址所在地，运行结果如图 2.85 所示。

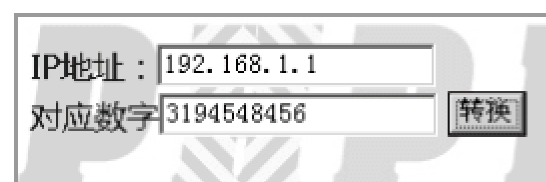


图 2.85　利用 IP 地址查找主机所在地

关键技术

首先需要验证输入的 IP 地址是否合法，如果合法，再将其转换为对应的数值。验证 IP 地址是否合法可以通过正则表达式实现。

IP 地址使用 4 位地址定位设备，它采用点分十进制数的格式“xxx.xxx.xxx.xxx”，例如 192.168.1.66。因此，验证 IP 地址是否合法的正则表达式如下：

```
/(\d+).(\d+).(\d+).(\d+)/
```

将 IP 地址转换为对应的数值可以应用以下公式实现：

```
P1*255*255*255+P2*255*255+P3*255+P4*1
```

在上面的公式中，P1 代表 IP 地址第 1 段的数值；P2 代表 IP 地址第 2 段的数值；P3 代表 IP 地址第 3 段的数值；P4 代表 IP 地址第 4 段的数值。

设计过程

（1）编写 JavaScript 脚本，定义 deal()方法验证用户提交的 IP 地址格式是否正确，定义 convert()方法将 IP 地址转换为对应数值，该方法只有一个参数 ip，用于指定 IP 地址，返回值为 IP 地址对应的数值，代码如下：

```
<script language="javascript">
	function deal(myform){                                          //自定义函数
		var ip=form1.ip.value;                                      //定义变量
		objExp=/(\d+).(\d+).(\d+).(\d+)/;                           //正则表达式
		if(ip==""){                                                 //条件判断
			alert("请输入 IP 地址!");form1.ip.focus();return;
	}
	if(objExp.test(ip)){
			myform.ipNum.value=convert(ip);
		}else{
			alert("您输入的 IP 地址不合法!");form1.ip.focus();return;
		}
}
	function convert(ip){                                           //自定义函数
		ip=ip.split(".");
		vip=ip[0]*255*255*255+ip[1]*255*255+ip[2]*255+ip[3]*1;      //转换程式
		return vip;                                                 //返回值
	}
</script>
```

（2）创建 index.php 文件，通过 form 表单提交要验证的 IP 地址，调用 JavaScript 脚本中的方法验证 IP 地址格式和完成格式的转换操作。

秘笈心法

心法领悟 111：使用$_SERVER["REMOTE_ADDR"]的注意事项。

在 PHP 中，可以通过$_SERVER["REMOTE_ADDR"]获取客户机的 IP 地址，但并不能判断该 IP 地址的所在地。如果要判断 IP 地址的所在地就必须对其格式进行转换。

实例 112	解决用 substr()函数对中文字符串截取时出现乱码的问题 光盘位置：光盘\MR\02\112	高级 趣味指数：★★★★

实例说明

使用 substr()函数是按字节截取字符串的，在截取中文字符串时，由于一个汉字由两个字符组成，如果只截取 1 个字符就会出现乱码。本实例使用自定义函数解决对中文字符串截取时的乱码问题，运行结果如图 2.86 所示。

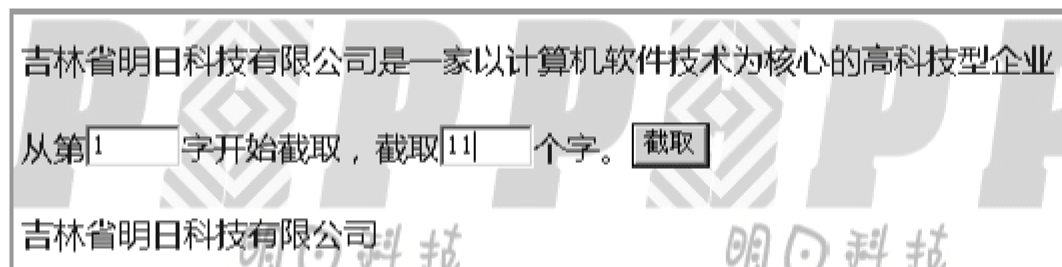


图 2.86　解决用 substr()函数对中文字符串截取时的乱码

关键技术

对中文字符串的截取虽说是通过自定义函数来完成的，但是其根本还是应用 substr()函数，只是在进行字符串截取时，对字符串的类型进行了判断。

对截取字符串中首个字节的 ASCII 序数值进行判断，如果 ASCII 序数值大于 0xa0，则表示为汉字，那么在

应用 substr()函数进行截取时，就以 2 个字节为单位；如果 ASCII 序数值小于 0xa0，则表示为英文字符串，那么在应用 substr()函数进行截取时，就以 1 个字节为单位。这样将中文字符串和英文字符串分隔进行截取就避免了出现乱码的问题。

自定义函数 msubstr()的语法如下：

```
function msubstr($str, $start, $len) {              //$str 指的是字符串，$start 指的是字符串的起始位置，$len 指的是长度
 $strlen = $start + $len;                           //用$strlen 存储字符串的总长度（从字符串的起始位置到字符串的总长度）
 for($i = 0; $i < $strlen; $i ++) {                 //通过 for 循环语句，循环读取字符串
        if (ord ( substr ( $str, $i, 1 ) ) > 0xa0) { //如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示为汉字
               $tmpstr .= substr ( $str, $i, 2 );    //每次取出两位字符赋给变量$tmpstr，即等于一个汉字
               $i ++; //变量自加 1
        } else {                                    //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
               $tmpstr .= substr ( $str, $i, 1 );
        }
 }
 return $tmpstr;                                    //输出字符串
}
```

其中，参数$str 是指定被截取的字符串；参数$strart 是截取的开始位置；参数$len 是截取的长度。返回值$tmpstr 是截取后的字符串。

设计过程

创建 index.php 文件，定义 msubstr()函数，对中文字符串进行截取。创建 form 表单，提交字符串截取的开始位置和长度，然后调用 msubstr()函数对字符串进行截取，并且输出截取结果，代码如下：

```
<?php
function msubstr($str, $start, $len) {              //$str 指的是字符串，$start 指的是字符串的起始位置，$len 指的是长度
 $strlen = $start + $len;                           //用$strlen 存储字符串的总长度（从字符串的起始位置到字符串的总长度）
 for($i = 0; $i < $strlen; $i ++) {                 //通过 for 循环语句，循环读取字符串
        if (ord ( substr ( $str, $i, 1 ) ) > 0xa0) { //如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示为汉字
               $tmpstr .= substr ( $str, $i, 2 );    //每次取出两位字符赋给变量$tmpstr，即等于一个汉字
               $i ++; //变量自加 1
        } else {                                    //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
               $tmpstr .= substr ( $str, $i, 1 );
        }
 }
 return $tmpstr;                                    //输出字符串
}
$string="吉林省明日科技有限公司是一家以计算机软件技术为核心的高科技型企业";
$str=msubstr($string,$_POST[te],$_POST[tx]);
echo $str;
?>
```

秘笈心法

心法领悟 112：利用函数截取中文字符串的注意事项。

使用 substr()截取中文最好首先定义算法，以免截取半个中文文字而出现乱码。

实例 113	字符串与 HTML 标记相互转换 光盘位置：光盘\MR\02\113	高级 趣味指数：★★★★

实例说明

本实例是通过 PHP 5.0 中的新型字符串实现字符串与 HTML 标记的相互转换，运行结果如图 2.87 所示。

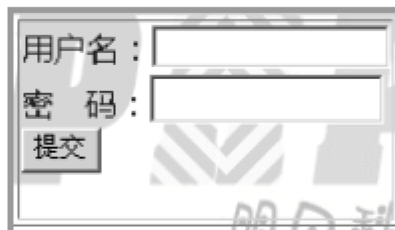


图 2.87　字符串与 HTML 标记的相互转换

关键技术

本实例的关键点是 PHP 5.0 新型字符串的使用，详细说明请参见实例 038。

设计过程

创建 index.php 文件，利用 PHP 5.0 新型字符串将 HTML 标记封装到 PHP 的变量中，并且输出 PHP 变量的值，代码如下：

```
<?php
$str=<<<mark                                        //新型字符串格式
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>字符串与 HTML 标记相互转换</title>
</head>

<body background="../image/1.jpg" style="font-size:16px; color:blue; font-family:'微软雅黑'"><table border="1"><tr><td>
 <form action="" method="post">
        用户名：<input type="text" name="test" size="15" /><br>
        密   码：<input type="password" name="pwd" size="15" /><br>
        <input type="submit" name="sub" value="提交" />
 </form></td></tr></table>
</body>
</html>
mark;                                               //新型字符串结束符
echo $str;
?>
```

秘笈心法

心法领悟 113：新型字符串的应用范围。

新型字符串被广泛应用在含有较多的 HTML 标记的 PHP 编码中。

实例 114	运用 PHP 5.0 新型字符串输出 XML 数据 光盘位置：光盘\MR\02\114	高级 趣味指数：★★★★

实例说明

本实例通过 PHP 5.0 新型字符串输出 XML 格式的数据，运行结果如图 2.88 所示。



图 2.88　PHP 5.0 新型字符串输出 XML 格式的数据

关键技术

本实例的关键点是 PHP 5.0 新型字符串的使用，详细说明请参见实例 038。

设计过程

创建 index.php 文件，定义 PHP 5.0 新型字符串，通过其输出 XML 格式的数据，代码如下：

```
<?php
$str=<<<mark
<?xml version="1.0" encoding="utf8"?>
<rss xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:taxo="http://purl.org/rss/1.0/modules/taxonomy/" version="2.0">
  <channel>
    <title>明日科技</title>
    <link>http://www.mingrisoft.com</link>
    <description>吉林省明日科技有限公司</description>
    <dc:creator>http://www.mingrisoft.com</dc:creator>
<item>
      <title>编程词典</title>
      <link>http://www.mrbccd.com</link>
      <description>明日科技编程词典重拳出击，隆重上市！</description>
      <pubDate>2008-08-08</pubDate>
      </item>
</channel></rss>
mark;
echo $str;
?>
```

秘笈心法

心法领悟 114：使用新型字符串的优点。

在 PHP 5.0 之前的开发中，想要在 PHP 中输出一些 html 标记是通过 echo 语句一行一行输出的，这无疑是一件非常头痛的事。现在 PHP 提供了新型字符串，相关的问题就可迎刃而解了。

实例 115 判断字符串中是否存在指定子串

光盘位置：光盘\MR\02\115

高级

趣味指数：★★★★

实例说明

判断一个字符串中是否含有另一个字符串的方法有很多，例如，正则表达式等。本实例通过 stristr()函数判断字符串中是否存在指定子串，运行结果如图 2.89 所示。



图 2.89　字符串中不存在子串

关键技术

判断一个字符串是否是另一个字符串的子串可通过函数 stristr()实现，其语法如下：

```
string stristr ( string haystack, string needle );
```

该函数获取指定字符串（A）在另一个字符串（B）中首次出现的位置到字符串（B）末尾的所有字符串。参数 haystack 指定查找的字符串，参数 needle 指定查找的对象。

该函数如果执行成功则返回剩余的字符串，否则将返回 FALSE。

设计过程

创建 index.php 文件，定义字符串变量，将该变量与通过 POST()方法传递进来的数据利用 stristr()函数进行比较并输出结果，代码如下：

```
<?php
    $a = "Hello World !!";                                              //定义字符串
    echo $a;                                                            //输出字符串
    echo "<form action="method='post'>";                                //输出表单
    echo "<input type='text' name='text' value='输入文本内容'>";
    echo "<input type='submit' name='sub' value='提交'>";
    echo "</form>";
    if($_POST[sub]){                                                    //通过 POST 方式传参
        if(stristr($a, $_POST[text]) != ""){                            //判断是否存在子串
            echo "<script>alert('文本存在指定子串');</script>";         //提示
        }else{
            echo   "<script>alert('文本不存在指定子串');</script>";     //提示
        }
    }
?>
```

秘笈心法

心法领悟 115：知识扩展。

stristr()函数还可以作为分割函数的一部分，从符合条件的字符串开始截取，一直到文件末尾。该函数不区分大小写，如果要在区分字母大小写的情况下进行搜索，可以应用 strstr()函数。

2.9　正则表达式

正则表达式是一种描述字符串结构模式的形式化表达方法，是一个强大、便捷、高效的文本处理工具，可用于验证用户输入的数据和检索大量的文本。在 PHP 中，表达式应用最好的体现是对表单提交的数据进行验证，判断是否合理、合法。

实例 116	验证电话号码的格式是否正确 光盘位置：光盘\MR\02\116	高级 趣味指数：★★★★

实例说明

表单注册时往往要求用户书写座机电话号码，而座机电话号码是由 11 位或 12 位数字组成的，所以一定要对电话号码的位数和格式进行限制。本实例通过正则表达式和正则表达式函数 preg_match()实现对电话号码格式的验证，运行结果如图 2.90 所示。



图 2.90　验证电话号码的格式是否正确

关键技术

（1）preg_match()：进行正则表达式匹配。

```
int preg_match ( string pattern, string subject [, array matches [, int flags]]);
```

（2）正则表达式：固定总位数为 11 位或 12 位的座机电话。

```
/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/;
```

设计过程

创建 index.php 文件，通过正则表达式函数对输入文本框的电话号码进行验证，并输出相关提示，其代码如下：

```
<?php
	if($_POST['sub']){
		if(preg_match("/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/",$_POST['text'])){		//正则表达式验证
			include("inc.php");								//包含信息提示页
			show_error("信息提示","格式正确","index.php");			//调用自定义函数输出提示信息
		}else{
			include("inc.php");
			show_error("信息提示","格式错误","index.php");			//调用自定义函数输出提示信息
		}
	}
?>
```

秘笈心法

心法领悟 116：使用正则表达式的注意事项。

正则表达式“/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/”是根据新式电话号码设置的，老式的座机电话是由 10 位数字组成的。

实例 117 验证 Email 地址格式是否正确

光盘位置：光盘\MR\02\117

高级

趣味指数：★★★★

实例说明

互联网发展到今天，几乎所有的 Web 爱好者都有自己的 Email 地址，无论申请的是 126 邮箱还是 163 邮箱，Email 地址的格式是固定的。本实例通过 preg_match()正则匹配函数和正则表达式验证 Email 地址格式是否正确，运行结果如图 2.91 所示。



图 2.91　验证 Email 地址格式是否正确

关键技术

（1）preg_match()函数：进行正则表达式匹配，详细参数参见实例 110。

（2）正则表达式：对 Email 地址进行验证。

```
/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/;
```

设计过程

创建 index.php 文件，当单击“验证”按钮时，通过正则表达式函数验证文本框传入的信息并输出对应提示，其代码如下：

```
if($_POST['sub']){
        require_once("inc.php");                                                    //包含信息提示页
        if(preg_match("/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/",trim($_POST['text']))){       //正则表达式验证
                show_error("信息提示","格式正确","index.php");                       //输出提示
        }else{
                show_error("信息提示","格式不正确","index.php");                     //输出提示
        }
}
?>
```

秘笈心法

心法领悟 117：PHP 支持的两种正则表达式函数库。

PHP 支持两种正则表达式函数库，一种是正则表达式函数库（POSIX 扩展），另外一种是正则表达式函数库（Perl 兼容）。在性能上，Perl 兼容正则表达式速度更快一些。

实例 118　验证 IP 地址是否有效

光盘位置：光盘\MR\02\118

高级

趣味指数：★★★★

实例说明

IP 地址是 Web 用户可以访问互联网的身份凭证。每一个 IP 地址相对其他用户的 IP 都是独立的。本实例通过正则表达式函数 preg_match()和正则表达式对 IP 地址进行验证，运行结果如图 2.92 所示。

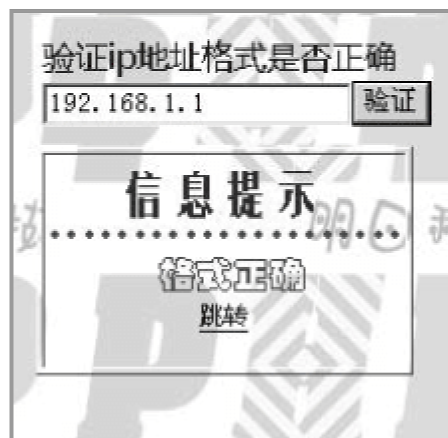


图 2.92　验证 IP 地址是否有效

关键技术

（1）preg_match_all()函数，在字符串 subject 中匹配表达式 pattern，函数返回匹配的次数。如果有数组 matches，那么每次匹配的结果将被存储到数组 matches 中，其语法如下：

```
int preg_match_all ( string pattern, string subject [, array matches] )
```

（2）正则表达式：对 IP 地址进行验证。

```
“/\d+\\d+\\d+\.\d+/”;
```

设计过程

（1）创建 index.php 文件，通过 form 表单提交电话号码，当单击“验证”按钮时，通过正则表达式函数验证 IP 地址，并且返回验证结果，其代码如下：

```
<?php
if($_POST['sub']){
```

```
    if(preg_match_all("/(\d{3}-)(\d{8})$|(\d{4}-)(\d{7})$/",$_POST['text']),$counts){              //验证电话号码
        include("inc.php");                                                                        //包含提示文件
        show_error("信息提示",$counts,"index.php");
    }else{
        include("inc.php");
        show_error("信息提示",$counts,"index.php");
    }
}
?>
```

（2）创建 inc.php 文件，定义方法 show_error()，根据传递的参数值定义返回的错误信息。

秘笈心法

心法领悟 118：IP 地址是什么。

Internet 为每一个主机分配唯一的一个 32 位地址，该地址称为 IP 地址，也称网际地址。IP 地址由 4 个数组成，每个数取值范围为 0~255，每两个数之间用“.”分隔。所以 IP 地址的格式是固定的，正则表达式“/\d+\\d+\\d+\.\d+/”可以验证所有 IP 地址。

实例 119 统计关键字的查询结果

光盘位置：光盘\MR\02\119

高级

趣味指数：★★★★

实例说明

统计关键字的查询结果的方法有很多，本实例通过正则表达式函数 split()和 count()实现统计关键字，运行结果如图 2.93 所示。

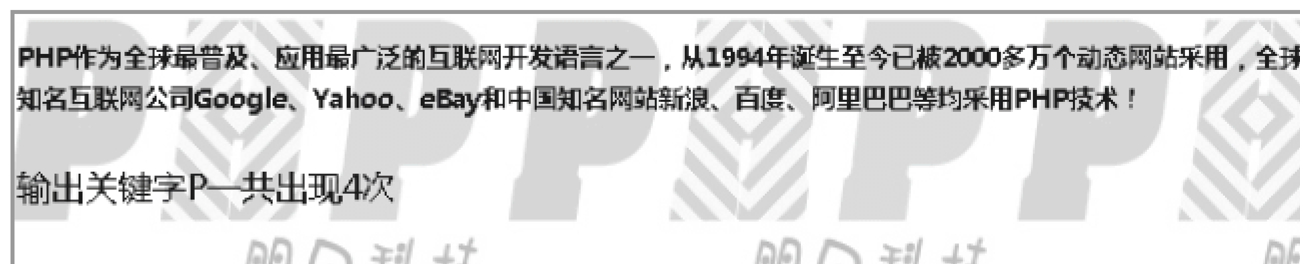


图 2.93 统计关键字的查询结果

关键技术

split()函数：用正则表达式将字符串分割到数组中。

```
array split ( string pattern, string string [, int limit]);
```

该函数返回一个字符串数组，每个单元为 string 经区分大小写的正则表达式 pattern 作为边界分割出的子串。如果设定了 limit，则返回的数组最多包含 limit 个单元，而其中最后一个单元包含 string 中剩余的所有部分。如果出错，则 split()返回 FALSE。

设计过程

创建 index.php 文件。首先，定义字符串变量$str 并为$str 赋值。然后，利用 split()函数检索字符“p”在$str 中的出现次数，并输出检索的结果，其代码如下：

```
<?php
                                                    //定义字符串变量
$str = "PHP 作为全球最普及、应用最广泛的互联网开发语言之一，从 1994 年诞生至今已被 2000 多万个动态网站采用，全球知名互联网公司
Google、Yahoo、eBay 和中国知名网站新浪、百度、阿里巴巴等均采用 PHP 技术！ ";
$b = split("P",$str);                               //使用 split()函数进行分割
echo "<h4 style ='red'>";
    echo $str."<br>";
    echo "</h4>";
```

```
    echo "输出关键字 P 一共出现".(count($b)-1)."次";                //利用 count()函数进行统计
?>
```

秘笈心法

心法领悟 119：split()函数的实现原理。

split()正则表达式函数是将字符串分割到数组中，如果字符串中有 n 个与 pattern 匹配的项目，则返回的数组将包含 n+1 个单元，所以 count()在计算时要减 1。

实例 120　计算查询操作的执行时间

光盘位置：光盘\MR\02\120

高级

趣味指数：★★★★

实例说明

计算机的每一步操作都是需要时间的，只不过由于时间过于短暂用户感觉不到。本实例通过时间戳函数 microtime()计算查询操作的执行时间，运行结果如图 2.94 所示。

id:1
查询操作所用时间为：0.004093秒

图 2.94　计算查询操作的执行时间

关键技术

获取查询操作的执行时间，在 PHP 中主要是使用 microtime()函数实现，其语法如下：

```
mixed microtime ( [bool get_as_float] );
```

该函数获取某一时刻的时间戳的微秒数。

设计过程

创建 index.php 文件。首先，连接数据库并设置页面的编码风格，在执行查询操作的开始与结束的两个时间点定义两个变量，分别利用 microtime()函数获取两点时间戳的微秒数。然后计算两个时间戳的差值，取得程序的执行时间，其代码如下：

```
<?php
    $conn = mysql_connect("localhost","root","111")or die("连接 mysql 出现错误");        //连接 mysql
    mysql_select_db("db_database02",$conn);                                                //连接数据库
    $sql = "SELECT * FROM tb_select";                                                      //sql 语句
    $a = microtime();                                                                      //取得当前时间戳的微秒数
    $rs = mysql_query($sql);                                                               //查询操作
    while($rst = mysql_fetch_array($rs))
    {
        echo "id:".$rst[0]."<br>";
        }
    $b = microtime();                                                                      //取得当前时间戳的微秒数
    echo "查询操作所用时间为：".($b-$a)."秒";                                              //输出结果
?>
```

说明： 本实例所用的数据表没有足够多的数据，以至于执行速度太快几乎不需要时间，为了显示结果笔者人为让程序休眠 1 秒钟。

秘笈心法

心法领悟 120：知识扩展。

time()和 mktime()在不指定参数的情况下具有相同的作用，都是返回自从 UNIX 新纪元（格林威治时间 1970

年 1 月 1 日 00:00:00）到当前时间的秒数。而 microtime()函数只是单纯取得两个时间戳的微秒数。

实例 121 查询关键字描红

光盘位置：光盘\MR\02\121

高级

趣味指数：★★★★

实例说明

关键字描红技术在实例 088 中已经进行过讲解，本实例是通过正则表达式替换函数 preg_replace()实现查询关键字描红，运行结果如图 2.95 所示。

吉林省长春市 搜索

用户名	性别	住址	QQ
杨明	男	吉林省长春市	555555555555
潘凯华	男	吉林省长春市	666666666
李中华	男	吉林省长春市	7777777777

图 2.95　查询关键字描红

关键技术

执行替换或关键字描红时经常使用 PHP 预定义函数 preg_replace()，其语法如下：

```
mixed preg_replace ( mixed pattern, mixed replacement, mixed subject [, int limit] );
```

参数说明：

mixed pattern：规定替换规则。

mixed replacement：替换部分。

mixed subject：替换内容的源信息。

主要功能：执行正则表达式的搜索和替换。

设计过程

创建 index.php 文件。首先，连接数据库并设置页面的编码风格。然后，当单击“搜索”按钮时，执行查询操作并将文本框输入的信息进行描红处理。最后，利用正则表达式函数 preg_replace()替换数据信息，其代码如下：

```
<?php
	$conn = mysql_connect("localhost","root","111")or die("连接 mysql 出现错误");		//连接 mysql
	mysql_select_db("db_database02",$conn);							//连接数据库
	mysql_query("SET NAMES GBK");								//设置编码格式
	if($_POST[sub]){									//通过 POST 方式传参
	//sql 语句
		$sql = "select * from tb_sel where tb_sel.name like ('%$_POST[text]%') or tb_sel.add like ('%$_POST[text]%') or tb_sel.sex like
('%$_POST[text]%') or tb_sel.QQ like('%$_POST[text]%')";
		$rs = mysql_query($sql);
		$a = $_POST[text];								//原参数
		$b = "<b><font color='#FF0000'>".$_POST[text]."</font></b>";			//描红后的参数
?>
	<table border="1" bordercolor="#0033CC" cellspacing="0"><tr><td>用户名</td><td>性别</td><td>住址</td><td>QQ</td></tr>
<?php
		while($rst = mysql_fetch_array($rs)){						//查询输出
			$rste = preg_replace("/$a/",$b,$rst);					//正则替换
											//输出表格
```

```
                echo "<tr><td>".$rste[name]."</td><td>".$rste[sex]."</td><td>".$rste[add]."</td><td>".$rste[QQ]."</td></tr><br>";
            }
        }
?>
</table>
```

秘笈心法

心法领悟 121：使用 preg_replace()函数的相关说明。

preg_replace()函数的每个参数（除了 limit）都可以是一个数组。如果 pattern 和 replacement 都是数组，将以其键名在数组中出现的顺序来进行处理。

实例 122　判断上传文件的类型

光盘位置：光盘\MR\02\122

高级

趣味指数：★★★★

实例说明

前面的实例中已经介绍了对文件类型的相关操作，本实例采用正则表达式函数 preg_match()实现判断上传文件的类型，运行结果如图 2.96 所示。



图 2.96　判断上传文件的类型

关键技术

使用 preg_match()函数进行正则表达式匹配，详细说明详见实例 110。

设计过程

创建 index.php 文件，当单击“上传”按钮时，利用 preg_match()正则表达式函数将上传的数据信息进行匹配并输出结果，其代码如下：

```
<?php
    if($_POST[sub]){                                        // if()条件语句
        if(preg_match("/.jpg/",strtolower($_POST[text]))){     //通过正则表达式函数判断类型
            echo "上传为图片类型";
        }else{
            if(preg_match("/.rar/",strtolower($_POST[text]))){
                echo "上传为压缩包类型";
            }else{
                echo "其他文件类型";
            }
        }
    }
?>
```

秘笈心法

心法领悟 122：知识扩展。

本实例使用的是 if…else…语句嵌套，在结构上看不是很明朗，读者可以运用 switch()函数改写此实例，相信在系统运行效率上，会有所提高。



实例 123 判断邮政编码格式是否正确

光盘位置：光盘\MR\02\123

高级

趣味指数：★★★★

用户在购买电子商务网站的商品时，商品一般都是以快递或邮寄的方式传递到用户手中，所以如果用户不能正确地填写邮寄地址或邮政编码，就有可能造成不必要的损失。本实例通过正则表达式函数 preg_match()验证用户提交的邮政编码格式是否正确，运行结果如图 2.97 所示。

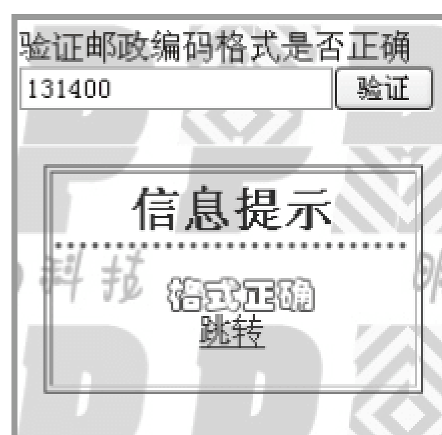


图 2.97　验证邮政编码是否有效

关键技术

（1）preg_match()函数：进行正则表达式匹配，详细参数参见实例 110。

（2）正则表达式：对邮政编码格式进行验证。

```
"[0-9]{6}";
```

设计过程

创建 index.php 文件，当单击“验证”按钮时，利用 preg_match()正则表达式函数对文本框的输入信息进行匹配并输出提示，其代码如下：

```
<?php
	if($_POST['sub']){                                              //通过 POST 方式传参
		require_once("inc.php");                                    //包含文件
		if(preg_match("/[0-9]{6}/",trim($_POST['text']))){          //正则表达式
			show_error("信息提示","格式正确","index.php");          //显示信息
		}else{
			show_error("信息提示","格式不正确","index.php");
		}
	}
?>
```

秘笈心法

心法领悟 123：什么是邮政编码。

邮政编码是为了实现邮政分拣自动化和邮政网络数字化设置的，以加快邮件邮递速度。目前世界上有 40 多个国家先后实行邮政编码制度，并以此作为衡量一个国家通信技术和邮政服务水平的标准之一。各国邮政编码规则并不统一。

2.10　数　　组

数组是对大量数据进行组织和管理的有效手段之一，通过数组的强大功能，可以对大量性质相同的数据进

行存储、排序、插入及删除等操作，从而可以有效地提高程序开发效率及改善程序的编写方式。

实例 124　创建并输出数组

光盘位置：光盘\MR\02\124

高级

趣味指数：★★★★

实例说明

如果只是想保存几个数据就没有必要和数据库交互，此时，可以选用数组。本实例通过数组函数 array()实现创建并输出数组，运行结果如图 2.98 所示。

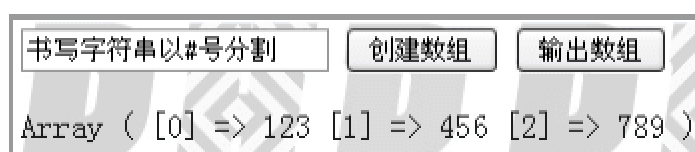


图 2.98　输出数组

关键技术

array()：新建一个数组。

```
array array ( [mixed ...] );
```

参数 mixed 的语法为“key=>value”，多个 mixed 参数之间用逗号隔开，分别定义了索引和值。索引可以是字符串或数字。如果省略了索引将是目前最大的整数索引+1。如果定义了两个完全一样的索引，则后面一个会覆盖前一个。数组中的各数据元素的数据类型可以不同，也可以是数组类型。当 mixed 是数组类型时，该数组就是二维数组。

设计过程

创建 index.php 文件，当单击“创建数组”按钮时，首先，判断文本框信息是否为空，如果为空便输出提示，否则将文本框信息保存在 SESSION 变量之中；当单击“输出数组”按钮时，首先，判断网页中是否存在 SESSION 变量，如果存在变量则利用字符串拆分函数进行拆分并打印输出，其代码如下：

```
<?php
	if($_POST[sub1]){                                                                          //通过 POST 方式传参
		if($_POST[text] == ""){                                                                //如果文本框为空
			echo "<script>alert('您创建了一个空数组');location.href='index.php'</script>";      //输出提示
		}else{
			$_SESSION['array'] = $_POST[text];                                                 //将结果保存在 SESSION 中
			echo "<script>alert('创建数组成功');location.href='index.php'</script>";            //提示
		}
	}
	if($_POST[sub2]){                                                                          //通过 post 方式传参
		if(isset($_SESSION['array'])){                                                         //如果 session 存在
			$array = explode("#",$_SESSION['array']);                                          //分割字符串
			print_r($array);                                                                   //打印
			session_destroy();                                                                 //销毁 session
		}else{
			echo "<script>alert('请先创建数组');location.href='index.php'</script>";            //提示
		}
	}
?>
```

秘笈心法

心法领悟 124：数组的重要作用。

数组是 PHP 基础语言中很重要的一部分，灵活运用数组会起到事半功倍的效果。

实例 125 统计数组元素个数

光盘位置：光盘\MR\02\125

高级

趣味指数：★★★★

实例说明

数组下标默认是以 0 为开始键值的整型数据，如果想取得数组元素的个数就要使用 count()函数。本实例通过 count()函数统计数组元素个数，运行结果如图 2.99 所示。

数组中存在元素4个

图 2.99 统计数组元素个数

关键技术

在 PHP 中，应用 count()函数对数组中的元素个数进行统计，其语法如下：

```
int count ( mixed array [, int mode])
```

参数说明：

array：必选参数，指定被统计的数组。

mode：可选参数，其参数值为 COUNT_RECURSIVE（或 1），如选中此参数，本函数将递归地对数组计数。对计算多维数组的所有单元尤其有用。此参数的默认值是 0。

设计过程

创建 index.php 文件，首先，定义数组变量并为此变量赋值。然后，利用 count()函数输出数组元素的个数，其代码如下：

```
<?php
$a = array('1','精细','asd','asd123');              //声明数组
echo "数组中存在元素".count($a)."个";                //计算数组元素个数
?>
```

秘笈心法

心法领悟 125：使用 count()函数的注意事项。

如果 count()函数的操作对象是 NULL，那么返回结果为 0。count()函数对没有初始化的变量返回 0，但对于空的数组也会返回 0。如果要判断变量是否初始化，则可以应用 isset()函数。另外，count()函数不能识别无限递归。

实例 126 向数组中添加元素

光盘位置：光盘\MR\02\126

高级

趣味指数：★★★★

实例说明

数组元素的个数和元素值是可以改变的。本实例通过数组函数 array_push()向数组中添加元素，运行结果如图 2.100 所示。

原数组：Array ([0] => 1 [1] => 精细 [2] => asd [3] => asd123)
添加元素后的数组：Array ([0] => 1 [1] => 精细 [2] => asd [3] => asd123 [4] => 哈哈)

图 2.100 向数组中添加元素

关键技术

在 PHP 中，向数组中添加元素使用 array_push()函数，其语法如下：

```
int array_push ( array array, mixed var [, mixed ...]);
```

参数说明：

array array：原数组。

mixed var：压入的数据元素。

该函数将 array 当成一个栈，并将传入的变量压入 array 的末尾。

设计过程

创建 index.php 文件，首先，定义数组变量，并为数组变量赋值。然后，利用 array_push()函数向数组中添加元素并打印数组输出信息，其代码如下：

```
<?php
	$a = array('1','精细','asd','asd123');		//声明一个数组
	echo "原数组：";
	print_r($a)."<br>";			//输出数组
	array_push($a,'哈哈');			//向数组末尾添加元素
	echo "<br>添加元素后的数组：";
	print_r($a);				//打印添加元素后的数组
?>
```

秘笈心法

心法领悟 126：使用 array_push()函数的相关说明。

如果用 array_push()函数单纯地给数组增加一个单元，不如选用$array[]=，因为这样没有调用函数的额外负担。

实例 127　将数组中指定索引位置的元素替换

光盘位置：光盘\MR\02\127

高级

趣味指数：★★★★

实例说明

创建数组时，由于疏忽将元素内容书写错误，更改错误的元素内容可以通过数组键值实现。本实例通过数组下标（键值）实现将数组中指定索引位置的元素替换，运行结果如图 2.101 所示。

原数组元素：Array ([0] => 1 [1] => 精细 [2] => asd [3] => asd123)
指定索引元素替换后的数组：Array ([0] => 1 [1] => 精细 [2] => 经济 [3] => asd123)
键值1的元素发生更改

图 2.101　将数组中指定索引位置的元素替换

关键技术

PHP 数字索引由数字组成，下标从 0 开始，数值索引一般表示数组元素在数组中的位置，数字索引数组默认索引值从数字 0 开始，不需要特别的指定，PHP 会自动为索引数组的键名赋一个整数值，然后从这个值开始自动增量，当然，也可以指定某个位置开始保存数据。

设计过程

创建 index.php 文件，定义数组变量并通过数组下标输出元素，其代码如下：

```
<?php
	echo "原数组元素：";
	$a = array('1','精细','asd','asd123');				//定义数组
	print_r($a);						//打印数组
	echo "<br>指定索引元素替换后的数组：";
	$a[2] = '经济';						//根据数组下标更改元素
	print_r($a);						//打印更改后的数组
	echo "<br>键值 2 的元素发生更改";
?>
```

秘笈心法

心法领悟 127：使用数组键值的注意事项。

当使用“组名称[键值]”这样的格式时，要注意数组键值是从 0 开始的，所以本例中更改键值为 2 的元素，其实是数组中的第 3 个元素。

实例 128 获取数组中最后一个元素

光盘位置：光盘\MR\02\128

高级

趣味指数：★★★★

实例说明

在数组元素比较多的情况下，想要取得数组中的最后一个元素可以通过先计算元素的个数再通过下标取得，还可以通过函数取得。本实例通过数组函数 array_pop()实现获取数组中的最后一个元素，运行结果如图 2.102 所示。

获取数组中最后一个元素是：i

图 2.102　获取数组中最后一个元素

关键技术

使用 array_pop()函数获取并返回 array 数组的最后一个单元，并将数组 array 的长度减 1，其语法如下：

```
mixed array_pop ( array array);
```

设计过程

创建 index.php 文件。首先，定义数组变量$a 并为此数组变量赋值。然后，利用函数 array_pop()弹出数组中的最后一个元素并输出结果，其代码如下：

```
<?php
	$a = array('a','b','c','d','e','f','g','h','i');			//声明数组
	echo "获取数组中最后一个元素是：".array_pop($a);		//返回数组中最后一个元素
?>
```

秘笈心法

心法领悟 128：使用 array_pop()函数的相关说明。

在使用 array_pop()函数时，数组中最后一个元素是被取出而不是被复制，即每次使用此函数，数组元素就会减少一个。如果数组为空（或者不是数组）将返回 NULL。

实例 129 去除数组中的重复元素

光盘位置：光盘\MR\02\129

高级

趣味指数：★★★★

实例说明

在数组中键值是唯一的，但是元素的值是可以重复的。想要删除数组元素中重复的值可以使用函数

array_unique()。本实例是通过数组函数 array_unique()去除数组中的重复元素值，运行结果如图 2.103 所示。

原数组元素为：Array ([0] => a [1] => b [2] => a [3] => d [4] => a [5] => f [6] => a [7] => h [8] => a)
去除重复项后的数组为：Array ([0] => a [1] => b [3] => d [5] => f [7] => h)

图 2.103　去除数组中的重复元素

关键技术

在 PHP 中，通过 array_unique()函数去除数组中的重复元素，其语法如下：

```
array array_unique ( array array);
```

参数 array 为指定参数的数组，其返回值为一个没有重复元素的新数组。

设计过程

创建 index.php 文件。首先，创建数组变量$a 并为其赋值。然后，利用 array_unique()去除数组中的重复元素值。最后，打印数组，其代码如下：

```
<?php
	$a = array('a','b','a','d','a','f','a','h','a');			//定义数组
	echo "原数组元素为：";
	print_r($a);							//打印数组
	echo "<br>去除重复项后的数组为：";
	print_r(array_unique($a));					//输出去除重复项的数组
?>
```

秘笈心法

心法领悟 129：使用 array_unique()函数的原理。

使用 array_unique()函数的原理是先将值作为字符串排序，然后对每个值只保留第一个遇到的键名，而忽略所有后面的键名。但这并不意味着在未排序的数组中同一个值的第一个出现的键名会被保留。

实例 130　字符串与数组的转换

光盘位置：光盘\MR\02\130

高级

趣味指数：★★★★

实例说明

实例 109 介绍了字符串与数字之间的转换，下面来介绍字符串与数组之间的转换。本实例是通过强制类型转换实现字符串与数组的转换的，运行结果如图 2.104 所示。

字符串转换为数组：Array ([0] => 123)

数组转换为字符串：Array

图 2.104　字符串与数组的转换

关键技术

虽然 PHP 是弱类型语言，但有时仍然需要用到类型转换。PHP 中的类型转换和 C 语言一样，非常简单，在变量前加上用括号括起来的类型名称即可。允许转换的类型如表 2.7 所示。

表 2.7　类型强制转换

转换操作符	转 换 类 型	举　　例
(boolean)	转换成布尔型	(boolean)$num、(boolean)$str
(string)	转换成字符型	(string)$boo、(string)$flo
(integer)	转换成整型	(integer)$boo、(integer)$str
(float)	转换成浮点型	(float)$str、(float)$str
(array)	转换成数组	(array)$str
(object)	转换成对象	(object)$str

设计过程

创建 index.php 文件。首先，定义数组变量并为此变量赋值，然后，利用强制类型转换的方法将字符串类型转换为数组类型，其代码如下：

```
<?php
	$a = "123";                         //声明字符串变量
	$b = (array)$a;                     //将字符串转换为数组
	echo "字符串转换为数组：";
	print_r($b);                        //打印数组
	echo "<hr>";
	$c = array('1');                    //定义数组
	$d = (string)$c;                    //将数组转换为字符串
	echo"<br>数组转换为字符串：";
	echo $d;                            //输出字符串
?>
```

秘笈心法

心法领悟 130：将数组强制转换成字符串的注意事项。

将数组强制转换成字符串时，数组将被转换成字符串 "Array"，因此无法通过 echo()或者 print()函数来输出数组的内容。

实例 131　对数组元素进行随机排序

光盘位置：光盘\MR\02\131

高级

趣味指数：★★★★

实例说明

本实例通过 shuffle()函数实现对数组元素进行随机排序，运行结果如图 2.105 所示。

原数组元素顺序为：经济 a b 123 asd
经随机排序后：a 经济 asd 123 b

图 2.105　对数组元素进行随机排序

关键技术

在 PHP 中，将数组元素进行随机排序使用 shuffle()函数，其语法如下：

```
void shuffle ( array array);
```

使用该函数将数组打乱，随机排序。

设计过程

新建 index.php 文件。首先，定义数组变量，利用 srand()函数播下随机数发生器种子。然后，利用 shuffle()

函数将数组打乱。最后，输出数组元素，其代码如下：

```
<?php
$numbers = array('经济','a','b','123','asd');
echo "原数组元素顺序为：经济  a b 123 asd";
srand ((float)microtime()*1000000);
shuffle ($numbers);
echo "<br>经随机排序后：";
while (list (, $number) = each ($numbers)) {
            echo "$number ";
}
?>
```

秘笈心法

心法领悟 131：使用 shuffle()函数的注意事项。

在使用 shuffle()函数将数组打乱时，必须用 srand()函数播下本函数的随机数发生器的种子，否则得不到预期结果。

实例 132　随机抽取数组中元素

光盘位置：光盘\MR\02\132

高级

趣味指数：★★★★

实例说明

可以把随机抽取数组中的元素看成一个彩票抽奖的大摇箱，在原理上两者是没什么区别的，但是彩票抽奖的大摇箱可能存在一点点的偶然性，例如大摇箱里的球弹力不同等。本实例通过随机函数 rand()实现随机抽取数组中的元素，运行结果如图 2.106 所示。

数组：Array ([0] => 经济 [1] => a [2] => b [3] => 123 [4] => asd)
随机取得数组的元素是：b

图 2.106　随机抽取数组中元素

关键技术

应用 rand()函数获取随机数值，并将随机数值指向数组下标从而实现随机获取指定元素的值。rand()函数的语法如下：

```
int rand ( [int min, int max]);
```

参数说明：

min：随机数的最小值。

max：随机数的最大值。

设计过程

创建 PHP 脚本。首先，定义数组变量并为数组变量赋值。然后，利用 rand()函数获取 0~4 中的一个随机数，并将此随机数赋给数组变量作为数组变量的键值。最后，输出此数组变量，其代码如下：

```
<?php
    $numbers = array('经济','a','b','123','asd');              //定义数组
    echo "数组：";
    print_r($numbers);
    $rand = rand(0,4);                                          //创建一个 0~4 的随机数字
    echo "<br>随机取得数组的元素是：".$numbers[$rand];          //将随机数字作为键值获取数组元素
?>
```

秘笈心法

心法领悟 132：随机函数 rand()的相关说明。

在使用随机函数 rand()时，如果没有提供可选参数 min 和 max，rand()返回 0~RAND_MAX 之间的伪随机数。例如想得到 10~30（包括 10 和 30）之间的随机数，使用 rand(5, 15)。

实例 133 二维数组的输出

光盘位置：光盘\MR\02\133

高级

趣味指数：★★★★

实例说明

二维数组可以简单理解为数组元素中还包含一个数组。MySQL 图形化管理工具 phpMyAdmin 中存储的每个数据表其实就是一个二维数组。本实例实现二维数组中数据的输出，运行结果如图 2.107 所示。

输出二维数组$a : Array ([1] => Array ([0] => a [1] => b) [2] => Array ([0] => c [1] => d) [3] => Array ([0] => e [1] => f))
输出二维数组$a键值为1元素$a[1] : Array ([0] => a [1] => b)

图 2.107　二维数组的输出

关键技术

如果创建的数组中的元素的值都是变量，此数组称为一维数组；如果创建的数组的元素值仍然是数组，那么就称为二维数组。也就是说，一个数组的元素如果仍是一个一维数组，则称这个数组是二维数组。

设计过程

创建 index.php 文件。首先，定义二维数组变量$a 并为此变量赋值，然后，利用 print_r()函数实现二维数组的输出，其代码如下：

```
<?php
	echo '输出二维数组$a：';
	$a = array(1=>array('a','b'),2=>array('c','d'),3=>array('e','f'));		//创建一个二维数组
	print_r($a)."<br>";
	echo '<br>输出二维数组$a 键值为 1 元素$a[1]：';
	print_r($a[1]);								//打印二维数组键值为 1 的元素
?>
```

秘笈心法

心法领悟 133：对 PHP 中数组的理解。

PHP 中的数组实际上是一个有序图。该图是一种把 values 映射到 keys 的类型，此类型在很多方面做了优化，因此可以把它当成真正的数组来使用，或列表（矢量）、散列表（是图的一种实现）、字典、集合、栈、队列以及更多可能性。因为可以用另一个 PHP 数组作为值，也可以很容易地模拟树。

实例 134 获取数组当前的键名和值

光盘位置：光盘\MR\02\134

高级

趣味指数：★★★★

实例说明

每一个数组元素都是由键名和值两部分组成的。想要获取它们可以通过循环来实现。本实例通过 foreach()

语句获取数组当前的键名和值，运行结果如图 2.108 所示。

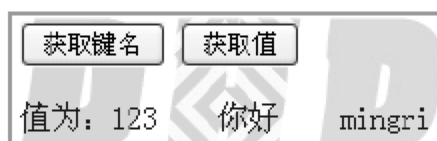


图 2.108　获取键名值

关键技术

数组中的每个实例都包含两项：键和值。其中键可以是数字、字符串或者数字和字符串的组合，用于标识数组中相应的值；而值被称为数组中的元素，可以定义为任意数据类型，甚至是混合类型，最终通过键来获取相应的值。

设计过程

新建 index.php 文件。首先，定义数组变量$array 并且为其赋值。当单击“获取键名”按钮时，利用 foreach()语句遍历数组并输出所有元素的键值；当单击“获取值”按钮时，同样通过 foreach()语句获取数组元素的 value 值并且输出，其代码如下：

```
<?php
    $array =array("编程"=>"123","soft"=>"你好","456"=>"mingri");          //定义数组
    if($_POST[sub]){                                                      //通过 POST 方式传参
        echo "键名为：";
        foreach($array as $key => $value){                                //遍历数组
            echo $key."    ";                         //输出键值
        }
    }
    if($_POST[sub1]){                                                     //通过 POST 方式传参
        echo "值为：";
        foreach($array as $key => $value){                                //遍历数组
            echo $value."    ";                       //输出值
        }
    }
?>
```

秘笈心法

心法领悟 134：使用 foreach()语句的注意事项。

foreach()语句只能用于数组。

实例 135	检测数组中是否存在某个值 光盘位置：光盘\MR\02\135	高级 趣味指数：★★★★

实例说明

PHP 大型的程序开发往往把程序员弄的焦头烂额，有时程序员不确定自己定义的数组中是否存在需要的元素。本实例通过 array_key_exists()函数检测数组中是否存在某个值，运行结果如图 2.109 所示。



图 2.109　检测数组中是否存在某元素

关键技术

在 PHP 中，检测数组中是否存在某个值，使用 array_key_exists()函数，其语法如下：

```
bool array_key_exists ( mixed key, array search );
```

参数说明：

mixed key：检测元素。

array search：源数组。

该函数检查给定的键名或索引是否存在于数组中。

设计过程

新建 index.php 文件，定义数组变量$array 并为其赋值，当单击“检测”按钮时，利用 array_key_exists()函数判断接收到的文本框信息是否存在于指定的数组元素中，并且根据返回值给出相应的提示信息，其代码如下：

```
<?php
	$array =array("编程"=>"123","soft"=>"你好","456"=>"mingri");			//定义数组
	if($_POST[sub]){										//通过 POST 方式传参
		if(array_key_exists($_POST[text],$array)){				//检测是否存在元素
			echo "<script>alert('数组中存在此元素');</script>";		//提示
		}else{
			echo "<script>alert('数组中不存在此元素');</script>";		//提示
		}
	}
?>
```

秘笈心法

心法领悟 135：array_key_exists()函数的相关说明。

array_key_exists()函数在给定的 key 存在于数组中时返回 TRUE，key 可以是任何能作为数组索引的值。array_key_exists() 也可用于对象。

实例 136	获取数组中的当前单元 光盘位置：光盘\MR\02\136	高级 趣味指数：★★★★

实例说明

本实例通过 current()函数获取数组中的当前单元，运行结果如图 2.110 所示。

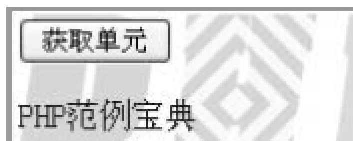


图 2.110　获取数组中的当前单元

关键技术

current()函数的主要功能是返回数组中的当前单元，其语法如下：

```
mixed current ( array &array );
```

设计过程

（1）创建 PHP 脚本文件。首先，定义数组变量并为其赋值，当单击“获取单元”按钮时，通过函数 current()获取数组当前元素，其代码如下：

```
<?php
	$array =array("编程"=>"PHP 范例宝典","soft"=>"你好","456"=>"mingri");			//定义数组
```

```
        if($_POST[sub]){                                        //通过 POST 方式传参
                echo current($array);                           //获取当前数组中的单元
        }
?>
```

（2）将该文件存储于\MR\02\136\文件夹中，命名为 index.php。运行结果如图 2.110 所示。

秘笈心法

心法领悟 136：使用 current()函数的相关说明。

current()函数返回当前被内部指针指向的数组单元的值，并不移动指针。如果内部指针指向超出了单元列表的末端，current()返回 FALSE。如果数组包含有空的单元（0 或者""，空字符串），则本函数在碰到这个单元时也返回 FALSE。

实例 137　从数组中随机取出元素

光盘位置：光盘\MR\02\137

高级　趣味指数：★★★★

实例说明

通过 rand()函数可以随机抽取数组中的元素。下面再通过另外一个函数 array_rand()实现从数组中随机取出元素，运行结果如图 2.111 所示。



图 2.111　从数组中随机取出元素

关键技术

当用户想从数组中取出一个或多个单元时，使用 array_rand()函数是非常有用的，它与 rand()函数获取随机数组元素的原理大同小异，其语法如下：

```
mixed array_rand ( array input [, int num_req] );
```

参数说明：

array input：传入的数组。

int num_req：可选参数，定义输出数组元素的个数。

设计过程

（1）创建 PHP 脚本文件。首先，定义数组变量$array 并为其赋值，当单击“获取单元”按钮时，通过函数 array_rand()获取数组中的随机元素值，其代码如下：

```
<?php
        $array =array("编程"=>"PHP 范例宝典","soft"=>"明日科技","456"=>"mingri");      //定义数组
        if($_POST[sub]){                                                              //通过 POST 方式传参
                $rand_keys = array_rand($array, 1);                                   //获取数组的随机键值
                echo "获取随机元素：".$array[$rand_keys] ;                              //输出随机元素值
        }
?>
```

（2）将该文件存储于\MR\02\137\文件夹中，命名为 index.php。运行结果如图 2.111 所示。

秘笈心法

心法领悟 137：使用 array_rand()函数的相关说明。

array_rand()函数在想从数组中取出一个或多个随机的元素时相当有用。它接受 $array 作为输入数组和一个

可选的参数 num_req，指明想取出多少个元素。如果没有指定，默认为 1。如果只取出一个，array_rand() 返回一个随机元素的键名，否则就返回一个包含随机键名的数组。这样用户就可以随机从数组中取出键名和值。

实例 138 合并数组

光盘位置：光盘\MR\02\138

高级

趣味指数：★★★★

实例说明

数组与数组之间是彼此独立的，但是两个数组可以合并到一起成为一个新数组。本实例通过数组函数 array_merge()实现合并数组，运行结果如图 2.112 所示。

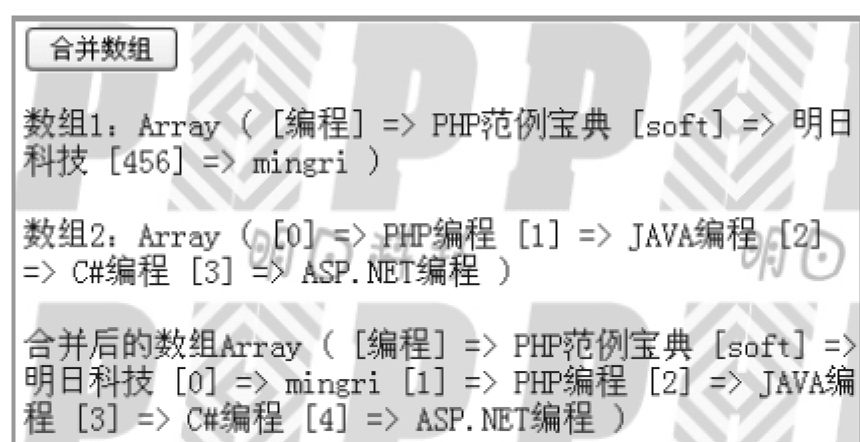


图 2.112　合并数组

关键技术

数组的合并应用的是 array_merge()函数，其语法如下：

```
array array_merge ( array array1 [, array array2 [, array ...]] );
```

参数说明：

array array1：数组 1。

array array2：数组 2。

设计过程

新建 index.php 文件。首先，定义两个数组变量并为其赋值。当单击“合并数组”按钮时，利用函数 array_merge()合并数组，并打印合并后的数组，其代码如下：

```
<?php
	$array =array("编程"=>"PHP 范例宝典","soft"=>"明日科技","456"=>"mingri");	//定义数组 1.
	echo "数组 1：";
	print_r($array);	//打印数组
	echo "<br><br>";
	$arr = array("PHP 编程","JAVA 编程","C#编程","ASP.NET 编程");	//定义数组 2.
	echo"数组 2：";
	print_r($arr);	//打印数组
	echo "<br><br>";
	if($_POST[sub]){	//通过 POST 方式传参
		echo "合并后的数组";
		print_r(array_merge($array,$arr));	//打印合并后的数组
	}
?>
```

秘笈心法

心法领悟 138：使用 array_merge()函数的相关说明。

array_merge()函数将一个或多个数组的单元合并起来，一个数组中的值附加在前一个数组的后面，返回作

为结果的数组。如果输入的数组中有相同的字符串键名，则该键名后面的值将覆盖前一个值。然而，如果数组包含数字键名，后面的值将不会覆盖原来的值，而是附加到后面。如果只给了一个数组并且该数组是数字索引的，则键名会以连续方式重新索引。

实例 139	拆分数组 光盘位置：光盘\MR\02\139	高级 趣味指数：★★★★

实例说明

存在数组的合并函数当然也就存在数组的拆分函数。本实例通过函数 array_chunk()实现数组的拆分，运行结果如图 2.113 所示。

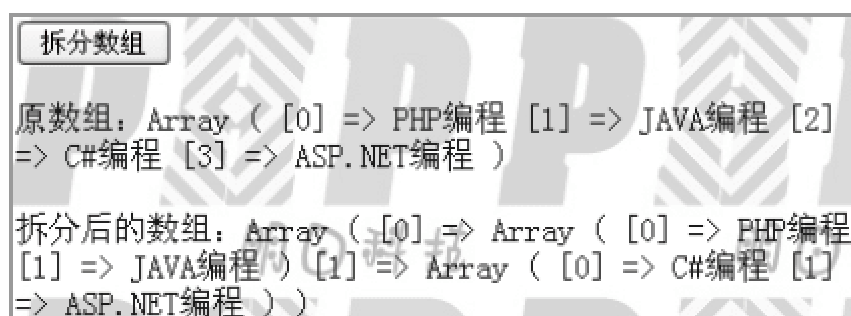


图 2.113　拆分数组

关键技术

array_chunk()函数可将一个数组分割成多个，其原理是将原有数据根据传递的参数不同分割成多个二维数组。array_chunk()函数的语法如下：

```
array array_chunk ( array input, int size [, bool preserve_keys] );
```

参数说明：

array input：源数组。

int size：分割成指定数量的数组。

设计过程

（1）创建 PHP 脚本文件。首先，定义数组变量并为其赋值，当单击“拆分数组”按钮时，打印被拆分后的数组，其代码如下：

```
<?php
	$arr = array("PHP 编程","JAVA 编程","C#编程","ASP.NET 编程");			//定义数组
	echo"原数组：";
	print_r($arr);									//打印数组
	echo "<br><br>";
	echo "拆分后的数组：";
	if($_POST[sub]){								//通过 POST 方式传参
		print_r(array_chunk($arr,2))."<br>";					//拆分数组
	}
?>
```

（2）将该文件存储于\MR\02\139\文件夹中，命名为 index.php。运行结果如图 2.113 所示。

秘笈心法

心法领悟 139：使用 array_chunk()函数的注意事项。

使用 array_chunk()函数拆分的数组，最后一个数组的单元数目可能会少几个。

实例 140 遍历数组 高级

光盘位置：光盘\MR\02\140 趣味指数：★★★★

实例说明

通过 for()语句可以遍历数组，下面通过另外一种方法来遍历数组。本实例通过 list()和 each()函数实现遍历数组，运行结果如图 2.114 所示。

```
遍历数组
Array ( [1] => PHP编程 [value] => PHP编程 [0] => 0 [key] => 0 )
Array ( [1] => JAVA编程 [value] => JAVA编程 [0] => 1 [key] => 1
) Array ( [1] => C#编程 [value] => C#编程 [0] => 2 [key] => 2 )
Array ( [1] => ASP.NET编程 [value] => ASP.NET编程 [0] => 3 [key]
=> 3 )
```

图 2.114 遍历数组

关键技术

遍历数组的方法有很多，除了使用 foreach()语句外，还可以通过 list()和 each()函数配合使用实现，其相关语法如下：

list()函数，把数组中的值赋给一些变量。与 array()函数类似，其不是真正的函数，而是语言结构。list()函数仅能用于数字索引的数组，且数字索引从 0 开始，其语法如下：

```
void list ( mixed ...)
```

参数 mixed 为被赋值的变量名称。

each()函数，返回数组中当前指针位置的键名和对应的值，并向前移动数组指针。键值对被返回为 4 个单元的数组，键名为 0、1、key 和 value。单元 0 和 key 包含有数组单元的键名，1 和 value 包含有数据，其语法如下：

```
array each ( array array)
```

设计过程

（1）创建 PHP 脚本文件，当单击“遍历数组”按钮时，首先，定义数组变量并为其赋值，然后，通过 list()和 each()函数实现数组遍历，并输出结果，其代码如下：

```
<?php
	if($_POST[sub]){							//通过 POST 方式传参
		$foo = array("PHP 编程","JAVA 编程","C#编程","ASP.NET 编程");	//定义数组
		for($a = 0;$a < count($foo);$a++){				//定义循环
			print_r(list($name, $value)=each($foo));		//打印数组
		}
}
?>
```

（2）将该文件存储于\MR\02\140 文件夹中，命名为 index.php。运行结果如图 2.114 所示。

秘笈心法

心法领悟 140：使用 list()和 each()函数的注意事项。

list()函数只能用于数字索引的数组。如果内部指针越过了数组的末端，则 each()函数返回 FALSE。

2.11　日期和时间

对日期和时间进行操作的程序无处不在，特别是在基于 Web 的应用程序中，如表单提交的时间、用户登录的时间、数据库中数据的更新和删除的时间等。想要记录这些操作执行的时刻，就需要通过日期和时间来完成。日期和时间在人们的生活中必不可少，同样，在互联网中也是非常重要的。

实例 141	设置系统的当前时间 光盘位置：光盘\MR\02\141	高级 趣味指数：★★★★

实例说明

系统的当前时间受时区限制，默认情况下是格林威治时间。用户正确取得本地时间是很重要的。本实例通过 date()函数获取系统的当前时间，运行结果如图 2.115 所示。



图 2.115　设置系统的当前时间

关键技术

因为在 PHP 语言中，日期、时间函数依赖于服务器的地区设置，而 PHP 默认设置的是标准的格林威治时间（即采用的是零时区），所以，如果没有对 PHP 的时区进行设置，那么使用日期、时间函数获取的将是英国伦敦本地时间（即零时区的时间）。

这也就是为什么要对 PHP 的时区进行设置的原因，因为如果不设置正确的时区，那么 PHP 的日期、时间函数就获取不到正确的当地时间。例如，以东八区为例，如果当地使用的是北京时间，那么如果没有对 PHP 的时区进行设置，获取的时间就将比北京时间少 8 个小时。

更改 PHP 语言中的时区设置如下有两种方法。

（1）在 php.ini 文件中，定位到[date]下的“;date.timezone =”选项，去掉前面的分号，并设置它的值为当地所在时区使用的时间。修改内容如图 2.116 所示。

例如，如果当地所在时区为东八区，那么就可以设置“date.timezone =”的值为 PRC、Asia/Hong_Kong、Asia/Shanghai（上海）或者 Asia/Urumqi（乌鲁木齐）等，这些都是东八区的时间。



图 2.116　设置 PHP 的时区

设置完成后，保存文件，重新启动 Apache 服务器。

（2）在应用程序中，在日期、时间函数之前使用 date_default_timezone_set()函数就可以完成对时区的设置。date_default_timezone_set()函数的语法如下：

```
date_default_timezone_set(timezone);
```

参数 timezone 为 PHP 可识别的时区名称，如果无法识别，则系统采用 UTC 时区。

例如，设置北京时间可以使用的时区包括 PRC（中华人民共和国）、Asia/Chongqing（重庆）、Asia/Shanghai（上海）和 Asia/Urumqi（乌鲁木齐），这几个时区名称是等效的。

设计过程

新建 index.php 文件，当单击“地区时间”按钮时，首先为当前 PHP 服务时间设置时区，并通过 switch()语句根据地址栏传递参数的不同，通过 date()函数获取不同国家和地区的时间，其代码如下：

```
<?php
	if($_POST[sub]){								//通过 POST 方式传参
		date_default_timezone_set("Asia/ShangHai");		//设置时区
		switch($_POST[select]){						//条件语句
			case "中国";							//如果是中国
			echo date("H:i:s");						//输出时间
			break;								//结束语句
			case "英国";							//如果是英国
			echo (date("H")-8).":".date("i:s");			//输出时间
			break;								//结束语句
			case "日本";							//如果是日本
			echo(date("H")+1).":".date("i:s");			//输出时间
			break;								//结束语句
			case "开罗";							//如果是开罗
			echo(date("H")-6).":".date("i:s");			//输出时间
			break;								//结束语句
			case "莫斯科";							//如果是莫斯科
			echo(date("H")-5).":".date("i:s");			//输出时间
			break;								//结束语句
		}
	}
?>
```

秘笈心法

心法领悟 141：知识扩展。

本实例只是简单地向用户介绍时区及不同地区相差的小时数，真正的开发中，不能对时间单纯地进行算术运算，需要使用关键技术中提供的方法对 PHP 的时区进行设置。

实例 142 将日期和时间转换为时间戳

光盘位置：光盘\MR\02\142

高级

趣味指数：★★★★

实例说明

将日期转换为时间戳是将时间进行数学运算的重要手段。本实例通过 time()函数实现将日期和时间转换为时间戳，运行结果如图 2.117 所示。

当前时间：2010-06-26 10:16:16
转换成时间戳为：1277518576

图 2.117　将日期和时间转换为时间戳

关键技术

获取系统 UNIX 时间戳的方法有很多，使用 time()函数是比较常用的一种方法，time()函数的语法如下：

```
int time ( void );
```

该函数没有参数，返回值为 UNIX 时间戳的整数值。

设计过程

创建 index.php 文件。首先，设置 PHP 的时区为中国上海。然后，利用 time()函数获取当前的时间戳，并通过 echo 语句输出时间戳，其代码如下：

```
<?php
date_default_timezone_set("Asia/ShangHai");                    //设置时区
echo "当前时间：".date('Y-m-d H:i:s');                          //取得当前时间
echo "<br>转换成时间戳为：".time();                             //转换为时间戳
?>
```

秘笈心法

心法领悟 142：知识扩展。

将当前日期和时间转换为时间戳的方法有很多，下面通过 strtotime()函数实现本例功能，代码如下：

```
<?php
date_default_timezone_set("Asia/ShangHai");                    //设置时区
echo "当前时间：".date('Y-m-d H:i:s');                          //设置当前日期和时间
echo "转换时间戳为：”strtotime("now");                          //转换时间戳
?>
```

实例 143　获取系统中的特定日期和时间

光盘位置：光盘\MR\02\143

高级

趣味指数：★★★★

实例说明

在 PHP 程序设计中，日期和时间是具有一定格式的，例如，今天是一年的第多少天等。本实例应用 date()函数通过传入参数不同实现日期和时间的格式化，运行结果如图 2.118 所示。

```
当前时间：2010年06月26日 10点21分28秒 AM
本月有30天.
今天是一周的第6天,是一年的第176天
```

图 2.118　日期和时间的格式化

关键技术

本实例通过 date()函数实现日期和时间的格式化，详细说明请参见实例 035。

设计过程

（1）创建 PHP 脚本文件，设置 PHP 的时区为中国上海。然后，利用向 date()函数中传递的参数不同实现对日期和时间的格式化，并输出格式化后的信息，其代码如下：

```
<?php
date_default_timezone_set("Asia/ShangHai");                    //设置时区
echo "当前时间：".date('Y 年 m 月 d 日  H 点 i 分 s 秒  A')."<br>";     //格式化日期和时间
echo "本月有".date('t')."天.<br>今天是一周的第".date('w')."天,是一年的第".date('z')."天";
?>
```

（2）将该文件存储于\MR\02\143\文件夹下，命名为 index.php。运行结果如图 2.118 所示。

秘笈心法

心法领悟 143：知识扩展。

date()函数还有很多其他的参数，熟悉这些参数的使用可以在使用格式化时间、日期上事半功倍。具体参数

说明，请参见实例 035。

实例 144　比较时间的大小

光盘位置：光盘\MR\02\144

高级

趣味指数：★★★★

实例说明

在 PHP 语言中，要完成对两个时间大小的比较，必须先将时间转换为时间戳，可以通过 strtotime()函数完成。本实例就应用 strtotime()函数对两个固定的时间进行比较，其运行结果如图 2.119 所示。



图 2.119　比较时间的大小

关键技术

PHP 中应用 strtotime()函数将任何英文文本的日期解析为 UNIX 时间戳，其值为相对于 now 参数给出的时间，如果没有提供此参数则用系统当前时间，此时与实例 142 中的 time()函数相同。strtotime()函数的语法如下：

```
int strtotime ( string time [, int now] );
```

该函数有两个参数。如果参数 time 的格式是绝对时间，则 now 参数不起作用；如果参数 time 的格式是相对时间，其对应的时间就是参数 now 来提供的，当没有提供参数 now 时，对应的时间就为当前时间。如果解析失败，则返回 false。在 PHP 5.1.0 之前的版本中，本函数在失败时返回-1。

设计过程

（1）创建 PHP 脚本文件。首先，编写<form>表单并设置当前时区为中国上海，当单击“日期比较”按钮时，运用 strtotime()函数获取两个不同时间点的时间戳，并进行求差运算。最后，输出相关提示，其代码如下：

```
<?php
	echo "<form action='' method='post'>";						//输出表单
	echo "<input type='submit' name='sub' value='日期比较'>";
	echo "</form>";
	date_default_timezone_set("Asia/ShangHai");				//设置时区
	if($_POST[sub]){										//通过 POST 传参
		$time = date("Y 年 m 月 d 日");						//取得当前日期
		$time_1 = time();								//取得当前时间戳
		$time_2 = strtotime("27 June 2010");					//取得固定日期的时间戳
		if($time_2 < $time_1){							//时间戳比较
			echo $time."比 2010 年 6 月 27 日大";
		}else{
			echo $time."比 2010 年 6 月 27 日小";
		}
	}
?>
```

（2）将该文件存储于\MR\02\144\文件夹下，命名为 index.php。运行结果如图 2.119 所示。

秘笈心法

心法领悟 144：使用 strtotime()函数的相关说明。

strtotime()函数预期接受一个包含美国英语日期格式的字符串，并尝试将其解析为 UNIX 时间戳（自 January 1 1970 00:00:00 GMT 起的秒数），其值相对于 now 参数给出的时间，如果没有提供此参数则用系统当前时间。

实例 145　计算考试时间

光盘位置：光盘\MR\02\145

高级

趣味指数：★★★★

实例说明

计算考试时间程序是单纯地将两个时间点的时间戳做算术运算。本实例通过time()函数实现考试时间的计算，运行结果如图 2.120 所示。



图 2.120　计算考试时间

关键技术

本实例的关键点是 time()函数的灵活运用。当单击“开始答题”按钮时，将 time()函数获取的时间戳存储到 SESSION 变量中；当单击“答题完毕”按钮时，再次通过 time()函数获取系统的当前时间戳。当前时间戳减去开始定义的时间戳，就是考试所用时间。

设计过程

（1）创建 PHP 脚本文件。首先，编写<form>表单并设置两个提交按钮，当每个按钮被单击后，利用 SESSION 变量保存此时刻的时间戳，并进行求差运算，然后输出结果，其代码如下：

```
<?php
    echo "<form action ='' method ='post'>";                                    //输出表单
    echo "<input type = 'submit' name='sub' value='开始答题'> <input type='submit' name= 'sub_1'value ='答题完毕'>";
    echo "</form>";
    if($_POST[sub]){                                                            //通过 POST 方式传参
        $time = time();                                                         //取得当前时间戳
        $_SESSION['time'] = $time;                                              //将时间戳保存在 SESSION 中
        echo "<script>alert('单击确定开始答题');</script>";                      //输出提示
    }
    if($_POST[sub_1]){                                                          //通过 POST 方式传参
        if(isset($_SESSION['time'])){                                           //判断 SESSION 是否存在
            echo "您用时".(time()-$_SESSION['time'])."秒答题";                   //输出结果
            session_destroy();                                                  //销毁 session
        }else{                                                                  //否则输出提示
            echo "<script>alert('您还没有答题无法结束，请先答题！');location.href='index.php'</script>";
        }
    }
?>
```

（2）将该文件存储于\MR\02\145\文件夹下，命名为 index.php。运行结果如图 2.120 所示。

秘笈心法

心法领悟 145：知识扩展。

本实例运用 SESSION 机制保存考试的开始时间，然后完成考试时间的计算。有关 SESSION 机制的详细讲解请参考本书第 5 章的内容。

实例 146　倒计时

光盘位置：光盘\MR\02\146

高级

趣味指数：★★★★

实例说明

倒计时功能的实现同样也是将时间戳做算术运算。本实例应用 strtotime()函数实现倒计时程序，运行结果如图 2.121 所示。

当前时间：2010年06月26日 11点25分50秒
距2011年元旦还有189天

图 2.121 倒计时

关键技术

本实例的关键点是 strtotime()函数的灵活运用，详细说明参见实例 144。

设计过程

（1）创建 PHP 脚本文件，设置当前时区为中国上海。首先，利用 time()函数取得当前时刻的时间戳并保存在变量中。然后，利用 strtotime()函数取得 2011 年 1 月 1 日的时间戳并保存在变量中，将两个时间戳进行求差运算，将结果转换成天数后输出，其代码如下：

```
<?php
	date_default_timezone_set("Asia/ShangHai");					//设置时区
	echo "当前时间：".date('Y 年 m 月 d 日  H 点 i 分 s 秒')."<br>";		//输出当前日期
	$time = time();									//取得当前时间戳
	$time_r = strtotime("1 January 2011");					//取得指定日期的时间戳
	echo "距 2011 年元旦还有<b style='color:red;'>".ceil((($time_r - $time)/(3600*24)))."</b>天";  //输出
?>
```

（2）将该文件存储于\MR\02\146\文件夹下，命名为 index.php。运行结果如图 2.121 所示。

秘笈心法

心法领悟 146：时间比较的原理。

通过上面几个实例用户应该明白，其实所有的时间比较归根结底都是时间戳的比较，所以掌握时间戳是十分重要的。

实例 147 网页闹钟

光盘位置：光盘\MR\02\147

高级

趣味指数：★★★★

实例说明

闹钟是人们生活中经常使用的一个小工具，它会在指定时间叫你起床。在 Web 程序中，也可以应用这个原理开发一个“网页闹钟”，提示用户在指定的时间或者时间段内要做什么工作。本实例通过 mktime()函数实现网页闹钟程序，运行结果如图 2.122 所示。

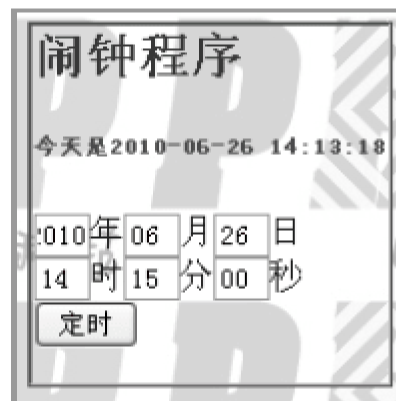


图 2.122 网页闹钟

关键技术

PHP 中，应用 mktime()函数将一个时间转换成 UNIX 的时间戳值。mktime()函数根据给出的参数返回 UNIX 时间戳。时间戳是一个长整数，包含从 UNIX 纪元到给定时间的秒数。mktime()函数的语法如下：

```
int mktime ( [int hour [, int minute [, int second [, int month [, int day [, int year [, int is_dst]]]]]]] );
```

mktime()函数的参数说明如表 2.8 所示。

表 2.8　mktime()函数的参数说明

参　数	说　明
hour	小时数
minute	分钟数
second	秒数（一分钟之内）
month	月份数
day	天数
year	年份数，可以是 2 位或 4 位数字，0-69 对应于 2000-2069，70-100 对应于 1970-2000。
is_dst	参数 is_dst 在夏令时可以被设为 1，如果不是则设为 0；如果不确定是否为夏令时，则设为–1（默认值）

设计过程

（1）创建 index.php 文件。首先，定义 PHP 时区为中国上海。然后，编写<form>表单，通过 mktime()函数实现获取当前的时间戳，其代码如下：

```
<?php
	echo "<table border='1' bordercolor='#FF0000' cellspacing='0' align='center'><tr><td>";	//输出表格
	echo "<h2 style='color:#0033FF'>闹钟程序</h2>";	//输出标题
	date_default_timezone_set("Asia/ShangHai");	//设置时区
	echo "<h6 style='color:red'>今天是".date("Y-m-d H:i:s")."</h6>";	//输出当前时间
	echo "<form action='' method='post'>";	//输出表单
	echo "<input type='text' name='text1' size='2'>年<input type='text' name='text2' size='2'>月";
	echo "<input type='text' name='text3' size='2'>日<br><input type='text' name='text4' size='2'>时";
	echo "<input type='text' name='text5' size='2'>分<input type='text' name='text6' size='2'>秒<br>";
	echo "<input type='submit' name='sub' value='定时'>";
	echo "</form>";
	$a = $_POST[text1];	//取得文本框内容
	$b = $_POST[text2];	//取得文本框内容
	$c = $_POST[text3];	//取得文本框内容
	$d = $_POST[text4];	//取得文本框内容
	$e = $_POST[text5];	//取得文本框内容
	$f = $_POST[text6];	//取得文本框内容
	$time = date(mktime($d, $e, $f, $b, $c, $a));	//获取时间戳
	$_SESSION['time'] = $time;	//保存在 SESSION 中
	if($_POST[sub]){	//通过 POST 传参
		echo"<script>window.location.href='in.php';</script>";	//跳转
	}
	echo "</td></tr></table>";
?>
```

（2）创建 in.php 文件，其代码如下：

```
<?php
	date_default_timezone_set("Asia/ShangHai");	//设置时区
	$time_1 = time();	//取得当前时间戳
	if($_SESSION['time'] < $time_1){	//判断条件
		echo "<script>alert('时间一去不复返');location.href='index.php'</script>";	//输出提示
	}else{
		if($_SESSION['time'] == $time_1){	//判断条件
			echo "<script>alert('今天是 XXX 生日');</script>";	//输出提示
		}else{	//输出提示
			echo "距离 XXX 的生日还有".ceil((($_SESSION['time']-$time_1)/(3600*24)))."天";
		}
	}
?>
```

秘笈心法

心法领悟 147：不同版本的操作平台对 UNIX 时间戳的支持。

PHP 为 UNIX 时间戳的处理提供各种函数。到目前的 PHP 版本为止，由于任何已知 Windows 版本以及一些其他系统均不支持负的时间戳，因此在 Windows 中无法表示 1970 年 1 月 1 日之前的时间。因为目前 UNIX 时间戳是以 32 位二进制表示的，其范围为（−2 147 483 648 ~ +2 147 483 647），因此，目前 UNIX 时间戳可表示的最大时间为 2038 年 1 月 19 日 3 点 14 分 7 秒，该时刻时间戳为 2 147 483 647，对于该时刻之后的时间，需要扩展表示 UNIX 时间戳的二进制位数。

实例 148 计算程序运行时间

光盘位置：光盘\MR\02\148

高级

趣味指数：★★★★☆

实例说明

在百度中，当执行一个查询操作时，获取到查询结果后，页面中就会出现一行文字，提示根据关键字搜索到多少个结果，以及搜索所用的时间。这个时间就是程序在执行该搜索时所用的时间，那么它是如何实现的呢？下面就模仿它来做一个计算程序运行时间的小程序，其运行结果如图 2.123 所示。

距离2010年上海世界博览会开幕还有 18 天!!!

倒计时程序的运行时间为 0.000308036804199 秒

图 2.123　计算程序运行时间

关键技术

本实例的关键点是 strtotime()函数的使用，其原理是：首先，在执行查询操作之前定义一个时间，将其精确到微秒，然后执行查询操作，接着，在查询功能执行完毕后，再次获取一个时间，同样精确到微秒，应用后获取的时间减去最初获取的时间，就是本次查询所用的时间。

设计过程

创建 index.php 文件。首先定义 run_time()方法，获取系统当前的时间戳。然后调用 run_time()方法，获取当前时间戳，接着运行程序。最后，再次调用 run_time()函数，获取系统当前时间，并且输出差值，获取程序的运行时间，其代码如下：

```
<?php
/*	声明 run_time 函数
通过 microtime()获取当前时间的微秒数和时间戳
应用 explode()函数将 microtime()函数返回的字符串进行分隔，返回一个数组
包括两个元素，一个元素是当前时间的微秒数，另一个是当前时间的时间戳
应用 list()函数将 explode()函数返回的数组值赋给指定的变量
最后，获取两个变量值的和
*/
function run_time(){
list($msec, $sec) = explode(" ", microtime());                  //使用 explode()函数返回两个变量
return ((float)$msec + (float)$sec);                           //返回两个变量的和
}
$start_time = run_time();                                       //第一次运行 run_time()函数
/*	运行 PHP 代码段	*/
    $time1 = strtotime(date( "Y-m-d"));                         //当前的系统时间
     $time2 = strtotime("2010-10-1");                           //上海世界博览会的开幕时间
     $sub2 = ceil(($time2 - $time1) / 86400);                   //(60 秒*60 分*24 小时)秒/天
  echo "距离 2010 年上海世界博览会开幕还有<font color=red>$sub2 </font>天!!!";
```

```
/*      *******************      */
$end_time = run_time();                                          //再次运行 run_time()函数
?>
<!--   输出差值，获取程序的运行时间  -->
倒计时程序的运行时间为<font color=blue> <?php echo ($end_time - $start_time); ?> </font>秒
?>
```

秘笈心法

心法领悟 148：在 PHP 中应用日期、时间函数时的注意事项。

☑ 在配置 PHP 时，进行正确的时区设置，这样就不会在输出时间时出现输出结果与当地时间不符的问题。

☑ 在正确地配置了 PHP 的时区之后，就是使用日期、时间函数获取日期、时间或者对应的时间戳，那么必须明确一个日期、时间在不同的操作系统中的有效范围，避免出现获取超出范围的日期、时间数据的现象。

☑ 有效的时间戳范围是格林威治时间 1901 年 12 月 13 日 20:45:54~2038 年 1 月 19 日 03:14:07（此范围符合 32 位有符号整数的最小值和最大值）。

☑ 在 Linux 系统中此范围限制为从 1901 年 12 月 13 日~2038 年 1 月 19 日。

☑ 在 Windows 系统中此范围限制为从 1970 年 1 月 1 日~2038 年 1 月 19 日。

☑ 在通过日期、时间函数获取日期、时间或者时间戳时，如果没有指定时间戳，那么获取的是本地时间。

第 3 章

Web 页面交互

- 获取表单元素的值
- 文本框组件的应用
- 下拉列表的应用
- 表单元素的动态操作
- CSS+ DIV 页面布局
- 弹出窗口
- 状态栏设计
- 导航菜单设计

3.1　获取表单元素的值

获取表单元素提交的值是表单应用中最基本的操作方法，通过获取表单元素提交的值可以实现很多的功能，例如，添加用户注册信息、提交用户登录信息、添加学生详细信息等。下面通过具体实例来介绍获取表单元素提交的值在 PHP 中的应用。

实例 149	设计论坛登录界面 光盘位置：光盘\MR\03\149	高级 趣味指数：★★★★

实例说明

论坛是许多网站不可或缺的模块之一，它为人们相互之间进行各种沟通和交流提供一个空间。设计论坛模块的第一个步骤就是设计论坛登录界面，让用户根据自己注册的用户名和密码登录论坛。在本例中笔者设计了一个简单的论坛登录界面，效果如图 3.1 所示。



图 3.1　登录界面

关键技术

在设计论坛登录界面时，主要采用表格标签进行界面布局，设计一个 3 行 2 列的表格，如图 3.2 所示。

用户名	
密　码	
确定	重置

图 3.2　利用表格标签布局

这样，一个论坛登录界面的轮廓就形成了。接下来还需要为表格设置背景图像，利用表格标签的 background 属性即可，代码如下：

```
<table width="392px" height="249px" align="center"><tr><td background="pic/tablebk.jpg">
```

设计过程

（1）利用 Dreamweaver 开发工具创建一个动态 PHP 页，将其保存为“index.php”。

（2）添加一个表单，在表单中插入一个 3 行 2 列的表格，并调整表格各行的高度和列的宽度。

（3）设置表格的 align 属性为“center”，使得表格在表单中始终居中显示。

（4）向表格单元格中输入用户名称和密码信息，并添加文本字段和按钮。

（5）设置“密码”文本字段的类型（type）属性为“password”，使其以掩码的形式隐藏真实密码。

（6）实例主要代码如下：

```
<form action="" method="post">
        <table align="center">
                <tr><td>用户名</td><td><input type="text" name="te" size="20" /></td></tr>
                <tr><td>密  码</td><td><input type="password" name="tex" size="22" /></td></tr>
                <tr><td align="left"><input class="two"type="submit" name="sub" value="       " /></td>
                <td align="left"><input class="three" type="reset" name="res" value="       " /></td></tr>
        </table>
</form>
```

秘笈心法

心法领悟 149：密码框与文本字段宽度不一致的解决方案。

许多初学人员都会遇到这种情况，在设计时密码框的宽度与文本字段宽度相同，但是在浏览器中显示则不相同。解决方法是在密码框和文本字段中指定统一的 CSS 样式，即设置 class 属性。在 PHP 页面中添加一个 CSS 样式：

```
<style type="text/css">
.fixwidth{
 width : 200px;
……}
</style>
```

这样，问题就解决了。

实例 150	可以上传图片的表单 光盘位置：光盘\MR\03\150	高级 趣味指数：★★★★

实例说明

许多大型网站为了吸引用户，会分配一些空间供用户使用。例如，大家经常使用的电子邮箱、QQ 个人空间等。为了能够让用户在个人空间中放置自己的内容，网站都会提供文件、图片等信息的上传功能。例如在 QQ 个人空间中，用户可以上传自己的照片，供好友欣赏。本实例利用文件域实现图片的上传功能，效果如图 3.3 所示。

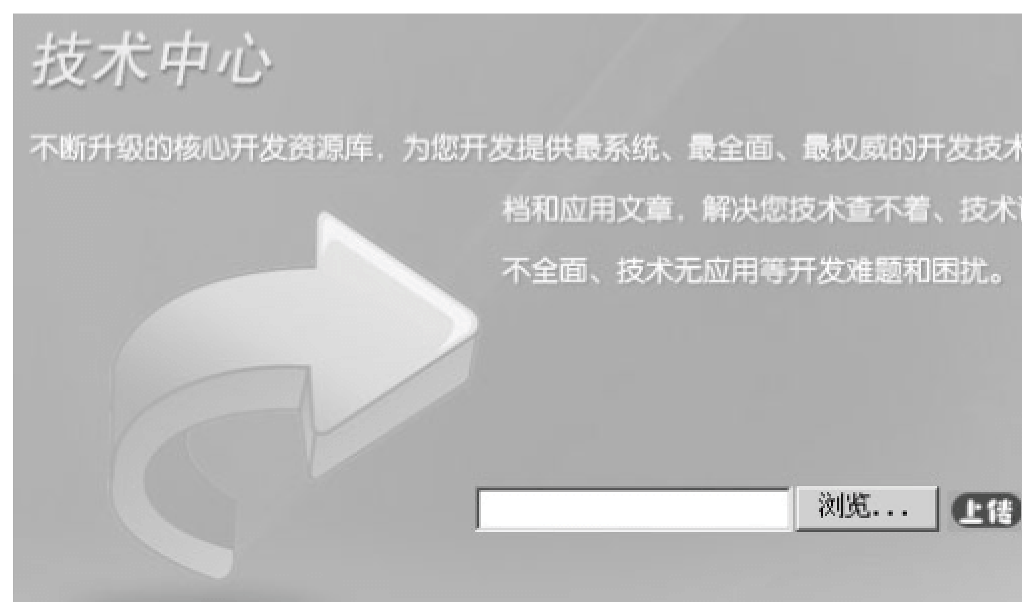


图 3.3　可以上传图片的表单

关键技术

在本例中，实现照片上传的主要方式是利用文件域让用户加载本地文件，然后应用 PHP 函数 move_uploaded_file()实现文件的上传，效果如图 3.4 所示。

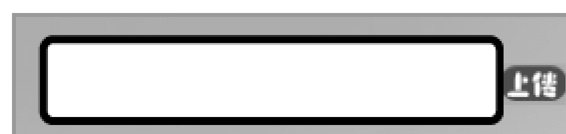


图 3.4　文本域加载本地文件

move_uploaded_file()函数可将上传的文件移动到新位置，其语法如下：

```
bool move_uploaded_file ( string filename, string destination );
```

参数说明：

string filename：指定的文件是合法的上传文件。

string destination：定义上传文件的名称。

设计过程

（1）创建动态 PHP 文件，命名为 index.php。

（2）添加一个表单，将 type 属性设置为 file，代码如下：

```
<form action="" method="post" enctype="multipart/form-data">
        <input type="file" name="upfile" size="20" />
        <input class="two"type="submit" name="sub" value="       " />
</form>
```

（3）利用<table>标签，编写网页框架。

（4）利用编码实现上传功能，核心代码如下：

```
<?php
if($_POST[sub]){                                                      //单击“上传”按钮
        if($_FILES['upfile']['name'] == ''){                          //判断上传文件名称是否为空
                echo "<script>alert('上传内容为空');</script>";
        }else{
                $info = $_FILES['upfile'];                            //将上传文件信息保存在数组中
                if($info['size'] > 0 && $info['size'] < 1024 * 8000){ //判断上传文件的大小
                        $dir = 'upfiles/';                            //定义变量
                        $name = $info['name'];                        //设定文件名称
                        $rand = rand(0,10000000);
                        $name = $rand.date('YmdHis').$name;
                        $path = 'upfiles/'.$name;
                        if(!is_dir($dir)){                            //判断文件夹是否存在，如果不存在则创建一个文件夹
                                mkdir($dir);
                        }
                        $move = move_uploaded_file($info['tmp_name'],$path);   //将上传文件移动到指定位置
                        if($move == true){
                                echo "<script>alert('上传文件成功');</script>";
                        }
                }else{
                        echo "<script>alert('上传文件过大');</script>";
                }
        }
}
?>
```

秘笈心法

心法领悟 150：解决上传文件重名问题。

如果用户建立个人网站，允许他人上传文件到服务器中，但是每一个浏览网站的用户并不知道服务器中是否存在与即将上传的文件重名的文件。大家都知道，在 Windows 下，同一个文件夹中，不允许两个同名文件的存在，这就使得用户上传的文件名称必须是独一无二的。笔者为上传文件的名称设置了如下代码：

```
$name = $info['name'];                                                //设定文件名称
$rand = rand(0,10000000);
$name = $rand.date('YmdHis').$name;
$path = 'upfiles/'.$name;
```

在一般情况下，是可以避免上传文件重名的现象的。



实例 151　通过 POST 方法提交表单元素　高级

光盘位置：光盘\MR\03\151　趣味指数：★★★★

实例说明

POST 方法是提交表单元素的方法之一，是程序员提交表单的主要途径。运行本实例如图 3.5 所示。



图 3.5　通过 POST 方法提交表单元素

关键技术

本实例实现通过 POST 方法提交表单元素，主要是通过定义<form>表单的“method”属性实现，定义方法如以下代码所示：

```
<form action="" method="post">
```

设计过程

（1）编写脚本文件 index.php。利用<table>标签建立网页框架，引入 CSS 文件。

（2）编写表单，将提交方法定义为 post，代码如下：

```
<form action="" method="post">                              //提交方法为 post
        <br>
        用户名：<input class="one" type="text" name="te" size="20" />
        <br>
        密  码：<input class="one" type="password" name="pwd" size="20" /><br>
        <input class="two" type="submit" name="sub" value="  " />
        <input class="three" type="reset" name="res" value="  " />
</form>
```

秘笈心法

心法领悟 151：为什么 POST 方法是程序员处理表单程序的首选？

POST 方法与 GET 方法不同，虽然都是采用地址栏传递参数的方式，但是通过 POST 方式传递的参数，在地址栏中是不可见，这样就大大降低了网站或者程序的外来因素的威胁。



实例 152　通过$_POST 方法获取表单元素　高级

光盘位置：光盘\MR\03\152　趣味指数：★★★★

实例说明

实例 151 介绍了如何通过 POST 方法提交表单元素，本实例讲解如何通过 POST 方法获取表单元素。运行

本实例如图 3.6 所示。

图 3.6　通过$_POST 获取表单元素

关键技术

本实例实现通过$_POST 方法获取表单元素，主要是利用预定义变量$_POST，该变量是一个自动全局变量，即它在所有的 PHP 脚本中都有效。

本实例通过单击按钮实现访问文本框信息的代码如下：

```
<tr>
<td width="100px" align="center"><?php echo $_POST[te];?></td>
<td width="100px" align="center"><?php echo $ POST[pwd];?></td>
</tr>
```

设计过程

（1）编写脚本文件 index.php。通过<table>标签编写网页框架，引入 CSS 文件，建立表单，代码如下：

```
<form action="" method="post">
        <br>
        用户名：<input class="one" type="text" name="te" size="20" />
        <br>
        密  码：<input class="one" type="password" name="pwd" size="20" /><br>
        <input class="two" type="submit" name="sub" value="  " />
        <input class="three" type="reset" name="res" value=" " />
</form>
```

（2）当单击“确定”按钮时，程序首先判断文本框和密码框信息是否为空，当不为空时，输出通过$_POST 方法获取的数据，核心代码如下：

```
<?php
        if($_POST[sub]){                                                    //单击按钮
                if($_POST[te] == "" || $_POST[pwd] == ""){                  //判断文本框和密码框是否为空
                        echo "<script>alert('文本框不能为空');</script>";
                }else{
?>
<table border="1" bordercolor="#0000CC" cellspacing="0" width="200px">
        <tr>
                <td align="center">用户名</td>
                <td align="center">密码</td>
        </tr>
                <tr>
                <td width="100px" align="center"><?php echo $_POST[te];?></td>
                <td width="100px" align="center"><?php echo $ POST[pwd];?></td>
        </tr>
</table>
<?php
                }
        }
?>
```

秘笈心法

心法领悟 152：超级全局变量$_POST。

虽然$_POST 是一个全局变量，但是在访问的时候不需要使用关键字 global。

实例 153　通过 GET 方法提交表单元素

光盘位置：光盘\MR\03\153

高级

趣味指数：★★★★

实例说明

GET 方法是提交表单元素的另一种方法，这种方法虽然不是程序员提交表单的首选，但是在日常的程序开发过程中也是很实用的。运行本实例如图 3.7 所示。

图 3.7　通过 GET 方法提交表单元素

关键技术

本实例实现通过 GET 方法提交表单元素，主要是通过定义<form>表单的“method”属性来实现，定义方法如以下代码所示：

```
<form action="" method="get">
```

设计过程

（1）编写脚本文件 index.php，利用<table>标签建立网页框架，引入 CSS 文件。

（2）编写表单，将提交方法定义为 get，代码如下：

```
<form action="" method="get">                                        //提交方法为 get
        <br>
        用户名：<input class="one" type="text" name="te" size="20" />
        <br>
        密  码：<input class="one" type="password" name="pwd" size="20" /><br>
        <input class="two" type="submit" name="sub" value="  " />
        <input class="three" type="reset" name="res" value="  " />
</form>
```

秘笈心法

心法领悟 153：GET 方法与 POST 方法的本质区别。

当用户提交表单时，观察 GET 方法和 POST 方法在地址栏的区别。POST 方法不会有任何变化，而 GET 方法将在地址栏内显示如图 3.8 所示的内容。

http://localhost/MR/03/005&006/in.php?te=mr&pwd=mrsoft&sub=%C2%A0%C2%A0%C2%A0%C2%A0%C2%A0%C2%A0%C2%A

图 3.8　GET 方法提交地址栏变化

虽然传递内容经过一些加密处理，但是也是近乎明文地将内容显示给别人，是非常不安全的行为。

实例 154　通过$_GET 方法获取表单元素

光盘位置：光盘\MR\03\154

高级

趣味指数：★★★★

实例说明

通过上面实例的学习，相信用户已经掌握了什么是 GET 方法、如何在表单中使用 GET 方法，下面将通过$_GET 方法向用户演示如何获取表单元素，如图 3.9 所示。



图 3.9　通过$_GET 方法获取表单元素

关键技术

本实例通过$_GET 方法获取表单元素，主要应用预定义变量$_GET，本变量是一个自动全局变量，即它在所有的 PHP 脚本中都有效。

本实例通过单击“确定”按钮，应用$_GET 方法获取文本框提交的数据，其代码如下：

```
<tr>
<td width="100px" align="center"><?php echo $_GET[te];?></td>
<td width="100px" align="center"><?php echo $_GET[pwd];?></td>
</tr>
```

设计过程

（1）编写脚本文件 index.php。通过<table>标签编写网页框架，引入 CSS 文件，建立表单，代码如下：

```
<form action="in.php" method="get">
        用户名：<input class="one" type="text" name="te" size="20" />
      <br>
        密  码：<input class="one" type="password" name="pwd" size="20" /><br>
        <input class="two" type="submit" name="sub" value=" " />
        <input class="three" type="reset" name="res" value=" " />
</form>
```

（2）当单击“确定”按钮时，程序首先判断文本框和密码框信息是否为空，当不为空时，输出通过$_GET 方法获取的数据，核心代码如下：

```
<?php
    if($_GET[sub]){                                          //单击按钮
        if($_GET[te] == "" || $_GET[pwd] == ""){             //判断文本框内容
            echo "<script>alert('文本框不能为空');</script>";
        }else{
?>
<table border="1" bordercolor="#0000CC" cellspacing="0" width="200px"><tr><td align="center">用户名</td><td align="center">密码</td></tr>
<tr><td width="100px" align="center"><?php echo $_GET[te];?></td><td width="100px" align="center"><?php echo
$_GET[pwd];?></td></tr></table>
<?php
```

```
            }
        }
?>
```

秘笈心法

心法领悟 154：超级全局变量$_GET。

与全局变量$_POST 一样，$_GET 方法页不需要使用关键字 global 来访问。通过$_GET 方法还可以获取到超链接传递的参数值。

3.2 文本框组件的应用

文本框组件是用来输入任何类型的单行文本、数字或字母数据的文本域。在进行程序设计过程中，经常使用文本框组件来获取用户输入的信息。下面通过具体实例来说明文本框组件在 PHP 中的应用。

实例 155	以文本域的形式显示数据信息 光盘位置：光盘\MR\03\155	高级 趣味指数：★★★★

实例说明

在开发网站的过程中，一般涉及到用户注册的网站模块都有一个用户注册的服务条款，个别的服务条款以整个网页进行显示，但是有的网站因服务条款篇幅过长，不适合使用这种方式，最好的解决方法是以文本域的形式来显示，既可节省页面，又美观大方。运行本实例，在文本域中显示用户注册的服务条款，运行效果如图 3.10 所示。

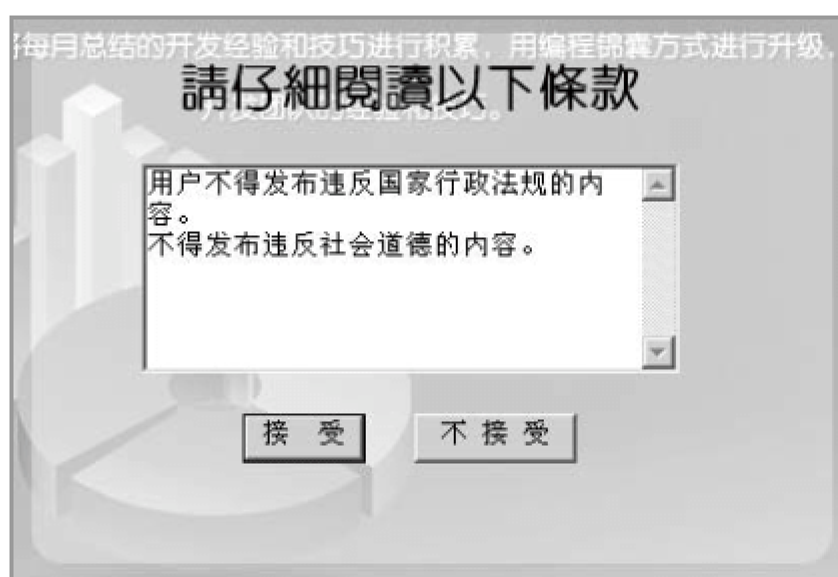


图 3.10 以文本域形式显示数据

关键技术

本实例主要应用表单中的文本域显示信息，主要就是设置文本域的初始值，代码如下：

```
<textarea name="test" cols="45" rows="8" id="test"><?php echo $myrow[text];?></textarea>
```

参数说明：

name：文本域的名称。

cols：文本域的字符宽度。

rows：文本域显示的行数。

设计过程

（1）编写脚本文件，编写文本域代码，并通过 PHP 变量传递初始值，代码如下：

```
<?php
$str = "用户不得发布违反国家行政法规的内容。                                    //定义变量
不得发布违反社会道德的内容。";
?>
<form action="" method="post">
<div align="center">
   <textarea name="txt" id="one" cols="30" rows="6"><?php echo $str;?></textarea>          //传递初始值
   <br>
   <br>
      <input type="submit" name="agreen" id="agreen" value="接　受(8)"   disabled="disabled"/>
            
      <input type="submit" name="disagreen" id="disagreen" value="不 接 受" />
     </div>
</form>
```

秘笈心法

心法领悟 155：文本域标签的初始值。

文本框标签和其他一些标签，设置初始值的方法是将初始内容写在“value”属性之中，而文本域标签的初始值内容是写在文本域的开始和结束标签之间的。

实例 156	限制多行文本域输入的字符个数 光盘位置：光盘\MR\03\156	高级 趣味指数：★★★★

实例说明

在网站的论坛或意见箱中，留言的内容通常都是在文本域中输入的，而文本域不具有限制用户输入最多字符数的功能，因此必须由开发人员编写代码进行控制。运行本实例，在“意见内容”文本框中只能输入 1000 个字节，即 1000 个英文字母或 500 个中文汉字，用户在文本框中每输入一个字符，下面的字节位置都将自动计算已输入字节数和剩余字节数，运行结果如图 3.11 所示。

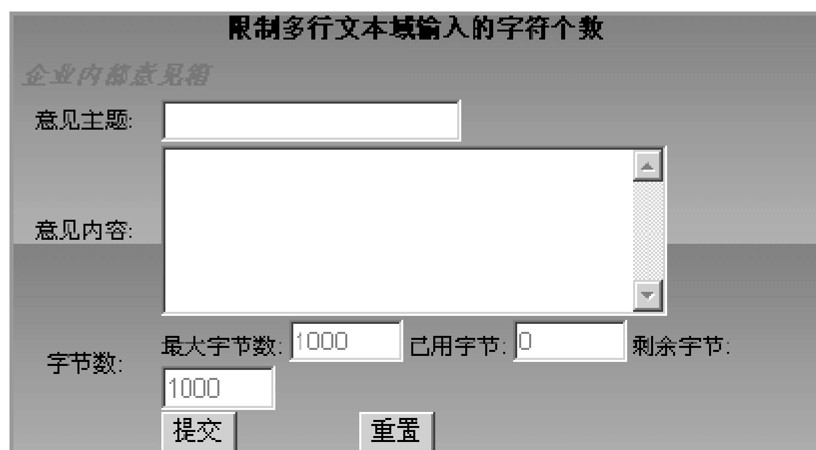


图 3.11　限制文本域输入的字符个数

关键技术

要实现限制多行文本域输入的字符个数，关键是应用 onKeyDown 事件和 onKeyUp 事件，调用自定义的 JavaScript 函数 StatByte()，来限制输入的最多字符数并计算已用字节数和剩余字节数。

设计过程

（1）创建一个企业内部意见箱的文本框，代码如下：

```
<textarea name="suggestions" cols="30" rows="5" class="wenbenkuang" id="suggestions"></textarea>
```

（2）添加显示字节数的文本框，包括最大字节数、已用字节和剩余字节，注意此时应该将这 3 个文本框设置为停用状态，不允许用户修改，代码如下：

```
最大字节数:
   <input name="maxs" type="text" disabled class="noborder" id="maxs"    value="1000" size="4">
```

```
已用字节:
<input name="used" type="text" disabled class="noborder" id="used"   value="0" size="4">
剩余字节:
<input name="remain" type="text" disabled class="noborder" id="remain" value="1000" size="4">
```

（3）编写自定义的 JavaScript 函数 StatByte()，用于判断用户输入的字符数是否大于允许输入的字符数，并计算已用字节数与剩余字节数，代码如下：

```
<SCRIPT language=JavaScript>
var LastCount =0;
function StatByte(Message,maxs,Used,Remain){                                    //字节统计
 var ByteCount = 0;
 var StrValue   = Message.value;
 var StrLength = Message.value.length;
 var maxsValue   = maxs.value;
if(LastCount != StrLength) {                                                     // 在此判断，减少循环次数
 for (i=0;i<StrLength;i++){
        ByteCount   = (StrValue.charCodeAt(i)<=256) ? ByteCount + 1 : ByteCount + 2;
        if (ByteCount>maxsValue) {
                Message.value = StrValue.substring(0,i);
        alert("留言内容最多不能超过 " +maxsValue+ " 个字节！ \n 注意：一个汉字为两字节。");
             ByteCount = maxsValue;
             break;
        }
}
    Used.value = ByteCount;
    Remain.value = maxsValue - ByteCount;
    LastCount = StrLength;
 }
}
</SCRIPT>
```

（4）修改“意见内容”文本域的代码，在 onKeyDown 事件和 onKeyUp 事件中调用自定义的 JavaScript 函数 StatByte()，限制输入的最多字符数并计算已用字节数和剩余字节数，代码如下：

```
<textarea name="suggestions" cols="30" rows="5" class="wenbenkuang" id="suggestions"
        onKeyDown="StatByte(this.form.suggestions,this.form.maxs,this.form.used,this.form.remain);"
        onKeyUp="StatByte(this.form.suggestions,this.form.maxs,this.form.used,this.form.remain);">
</textarea>>
```

秘笈心法

心法领悟 156：字节数的计算。

在计算字节数时，需要判断输入的字符是汉字还是英文字母或数字，因为一个汉字占两个字节。

实例 157　设置文本框的只读属性

光盘位置：光盘\MR\03\157

高级

趣味指数：★★★★

实例说明

在网页中，有些文本框中的信息是不允许浏览者进行修改的，即用户只有浏览的权限，而没有修改的权限。此功能可以通过设置文本框的只读属性来实现。运行本实例，在页面中修改库存数量时，将弹出“此文本框为只读属性，库存数量不能修改！！”的提示框，运行结果如图 3.12 所示。

关键技术

本实例的关键是文本框中 readonly 属性的运用，通过它可设置文本框的只读属性，然后通过 onfocus 事件调用 JavaScript 脚本，当用户单击已经设置为只读属性的文本框时，弹出一个提示对话框。

使用 readonly 属性实现只读的代码如下：

```
<input id="o" class="o" type="text" readonly value='0100521041'/>
```

图 3.12　设置文本框的只读属性

设计过程

新建 index.php 文件。编写表格，在表格中插入<form>标签，编写文本框，设置文本框的属性为 readonly，代码如下：

```
<form action="" method="post">
商品编号：<input id="o" class="o" type="text" readonly value='0100521041'/><br>
商品名称：<input id="o"type="text" value="酸牛奶" onfocus="this.value=""" /><br>
库存数量：<input id="o" class="n"type="text" disabled="disabled" value="100 件" /><br>
管理员工：<input id="o"type="text" value="001" />
</form>
```

秘笈心法

心法领悟 157：使用 readonly 的小技巧。

在使用 readonly 时，只需在<input>标记中加入属性名称即可。

实例 158	自动计算金额 光盘位置：光盘\MR\03\158	高级 趣味指数：★★★★

实例说明

在物流管理系统中，经常需要计算出库商品的金额或入库商品的金额，若以手工计算，比较麻烦而且容易出错，如果在程序中实现自动计算金额功能，则可以减少许多不必要的麻烦。运行本实例，如图 3.13 所示。

图 3.13　自动计算金额

关键技术

本实例主要通过触发表单控件的 Blur 事件，即表单控件失去焦点时，实现自动计算金额。实现此功能的 JavaScript 代码如下：

```
$(".o").blur(function(){                                    //当商品单价文本框失去焦点
        var price = $(".o").val();
        var num = $(".n").val();
        var amount = price * num;
        $(".e").val(amount);
});
```

设计过程

（1）编写脚本文件并命名为 index.php。引入 CSS 样式表文件，导入 JavaScript 脚本。编写表单，当商品单价失去焦点时自动计算金额，核心代码如下：

```
<form action="" method="post" name="form1">
商品编号：<input id="o" type="text" readonly value='酸牛奶'/><br>
商品单价：<input id="o" type="text"class="o" value="0.0"/><br>
库存数量：<input id="o" class="n"type="text" value="35" readonly /><br>
金额总计：<input id="o" class="e"type="text" value="0.0" />
</form>
```

秘笈心法

心法领悟 158：关于 Blur 事件。

用于当 Blur 事件(即失去焦点事件)发生时执行 JavaScript 代码。适用对象包括 Button、Checkbox、FileUpload、Layer、Password、Radio、Reset、Select、Submit、Text、Textarea 和 Window。

实例 159	为文本框设置默认值 光盘位置：光盘\MR\03\159	高级 趣味指数：★★★★

实例说明

在实例 157 中，笔者已经向用户介绍了文本框只读属性的相关知识。接下来向用户介绍一种与只读属性类似的属性，即设置文本框默认值。运行本实例，如图 3.14 所示，设置文本框的默认颜色为灰色，当在用户名中输入信息时动态改变文本框默认值信息。

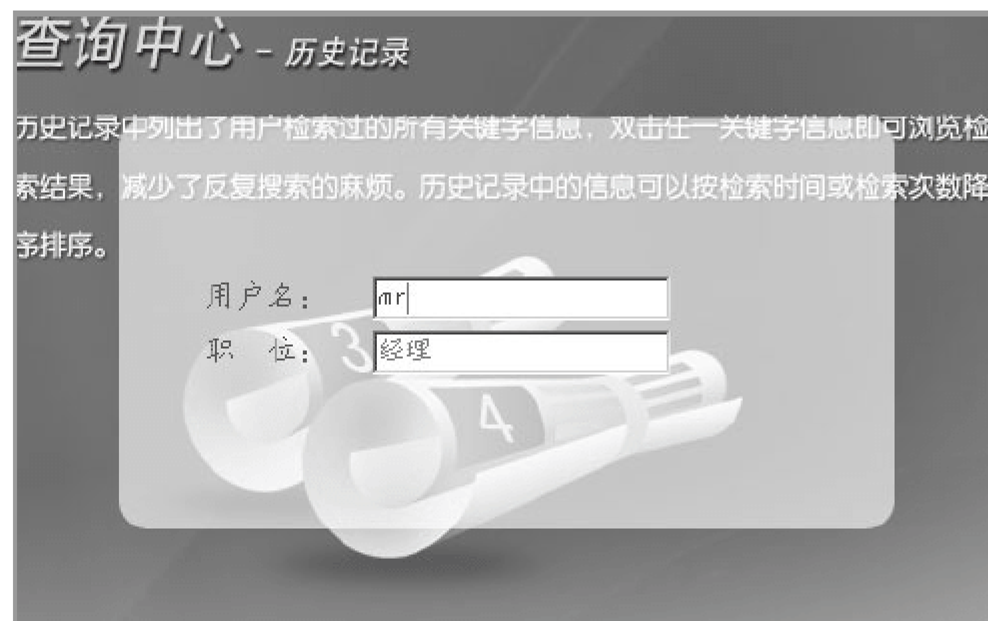


图 3.14　为文本框设置默认值

关键技术

本实例的关键是通过 disabled 属性设置文本框的默认值，然后通过相关事件，动态改变默认值文本框的值。

使用 disabled 属性设置默认值的代码如下：

```
<input id="o" class="o" type="text" disabled   value='0100521041'/>
```

设计过程

（1）新建 index.php 文件，引入 CSS 样式表文件，编写表单，代码如下：

```
table width="350" border="0" class="p">
  <tr>
    <td width="100">用户名：</td>
    <td align="left"><input class="m" id="name"type="text" /></td>
  </tr>
  <tr>
    <td>职  位：</td>
    <td align="left"><input class="m"type="text" id="type" disabled value="自动选取"/></td>
  </tr>
</table>
```

（2）定义 JavaScript 脚本方法，动态改变文本框的默认值，代码如下：

```
$("#name").keyup(function(){
        var name = $("#name").val();
        var manager = 'mr';
        if(name == 'mr'){
                $("#type").val("经理");
        }else{
                $("#type").val("程序员");
        }
});
```

秘笈心法

心法领悟 159：关于设置文本框的默认值。

（1）设置文本框的默认值与设置文本框的只读属性，在效果上是相同的。

（2）在使用 disabled 时，只需在<input>标记中加入属性名称即可。

实例 160	设置文本框的样式 光盘位置：光盘\MR\03\160	高级 趣味指数：★★★★

实例说明

<form>表单的文本框在默认情况下不够美观，使用默认的文本框样式将影响网站的视觉效果。运行本实例，如图 3.15 所示，利用 CSS 代码和简单的 JavaScript 代码，使文本框变得生动、有特色。



图 3.15　设置文本框样式

关键技术

本实例首先为文本框定义 CSS 样式，然后通过设置鼠标事件调用 CSS 样式，动态改变文本框的边框颜色。使用 CSS 样式的代码如下：

```
.mouse{
border:1px #FF0000 solid;
background-color:#FFCCFF;
}
```

使用 JavaScript 定义鼠标事件，代码如下：

```
$("#n").mouseover(function(){
        $("#n").css({border:"1px #0099FF solid",background:"white"});
        $("#n").mouseout(function(){
                $("#n").css({border:"1px red solid",background:"#FFCCFF"});
        });
});
$("#na").mouseover(function(){
        $("#na").css({border:"1px #0099FF solid",background:"white"});
        $("#na").mouseout(function(){
                $("#na").css({border:"1px red solid",background:"#FFCCFF"});
        });
});
```

设计过程

（1）新建 index.php 文件，建立网页架构，编写<form>表单，其关键代码如下：

```
<table width="350" border="0" class="p">
  <tr>
    <td width="100">用户名：</td>
    <td align="left"><input type="text" class="mouse" id="n"/></td>
  </tr>
  <tr>
    <td>密  码：</td>
    <td align="left"><input type="text" class="mouse" id="na" /></td>
  </tr>
</table>
```

（2）引入 CSS 样式和 JavaScript 脚本，核心代码如关键技术中所示。

秘笈心法

心法领悟 160：如何引入 CSS 样式。

定义 CSS 样式表后需要引入到相关脚本文件中才能起到作用。一般是在<meta>标签下，利用<link>标签引入。以本实例为例，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/inc.css"/>
```

实例 161	文本域的滚动条 光盘位置：光盘\MR\03\161	高级 趣味指数：★★★★

实例说明

文本域本身有自带的滚动条，用户可以通过滚动条浏览文本域中的所有内容，但是在初始情况下，用户可以根据需要设置文本域的列宽，也就是确定文本域在初始情况下显示多少内容，运行本实例如图 3.16 所示。

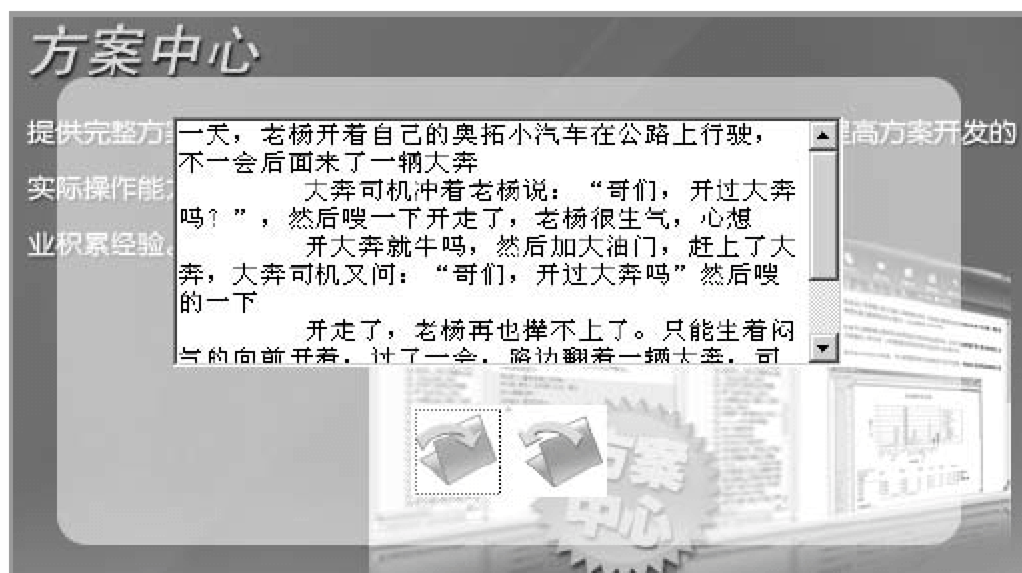


图 3.16　文本域的滚动条

关键技术

本实例的关键点是利用 JavaScript 代码，动态改变文本框的 rows 属性。JavaScript 代码如下：

```
$("#down").click(function(){
        $("textarea").attr("rows","8");
        $("#up").click(function(){
                $("textarea").attr("rows","1");
        });
});
```

设计过程

（1）新建 index.php 文件，通过<table>标签建立网页框架，输入文本域代码，引入 CSS 样式表和 JavaScript 脚本，代码如下：

```
<table width="600" height="600" border="0" align="center" background="pic/bg.jpg">
  <tr>
    <td align="center">
      <textarea cols="40"><?php echo $str;?></textarea><br><br>
      <input   type="button" id='down' value='  '/>
      <input   type="button" id='up' value='  '/>
</td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件，输入如关键技术所示代码。

秘笈心法

心法领悟 161：编写 JavaScript 代码的注意事项。

编写 JavaScript 代码时，要注意 JavaScript 代码是区分函数的大小写的。

3.3　下拉列表的应用

下拉列表是一种最节省空间的数据显示方式，正常状态下只能看到一个选项，单击控制按钮后，可以显示一定数量的选项，如果超出这个数量，会自动显示滚动条，浏览者可以通过拖动滚动条来查看各选项。

实例 162	省市级联动菜单 光盘位置：光盘\MR\03\162	高级 趣味指数：★★★★

实例说明

在进行网站开发的过程中，经常会用到相互关联的菜单。例如，省、市、县 3 级联动或者省、市 2 级联动

菜单。在本实例中，将讲解如何实现省、市 2 级联动菜单，其运行效果如图 3.17 所示。



图 3.17　省、市级联动菜单

关键技术

本实例通过设置 change 事件实现联动的下拉列表，并且根据 1 级下拉列表传递的不同参数，显示不同的 2 级下拉列表信息。实现这些功能的 JavaScript 代码如下：

```
$("#select1").change( function() {
        var id = $("#select1").val();
        if(id == 1){
                $.get('index.php', null, function(data){
                        $("span").empty();
                        $("span").append("<select><option>长春</opton><option>松原</opton><option>通辽</opton></select>");
                });
        }else{
                $.get('index.php', null, function(data){
                        $("span").empty();
                        $("span").append("<select><option>大连</opton><option>旅顺</opton><option>鲅鱼圈</opton></select>");
                });
        }
});
```

设计过程

（1）创建脚本文件，引入 CSS 样式表和 JavaScript 脚本。编写 1 级下拉列表，设置<option>标签属性“value”的值，代码如下：

```
<select id="select1">
        <option value="" selected>请选择</option>
        <option value="1">吉林</option>
        <option value="2">辽宁</option>
</select>
```

（2）编写 JavaScript 代码，当选择 1 级下拉列项时，触发 change 事件，代码如关键技术中所示。

秘笈心法

心法领悟 162：JavaScript 脚本语言的小技巧。

如果在 JavaScript 脚本中出现长语句，可以在 JavaScript 语句的最后加上字符“\”，使一条长语句分多行编写。

实例 163	省市县级联动菜单 光盘位置：光盘\MR\03\163	高级 趣味指数：★★★★

实例说明

实例 162 中讲解了如何实现 2 级联动菜单，下面将向用户讲解如何实现省、市、县 3 级联动菜单。运行本

实例，如图 3.18 所示。



图 3.18　省、市、县级联动菜单

关键技术

实现省、市、县 3 级联动菜单与实现 2 级联动菜单的思路大致相同，只不过需要套用多层的 if…else…语句实现。

设计过程

（1）创建脚本文件，命名为 index.php。创建 form 表单，编写下拉列表框，定义 2 级联动和 3 级联动的下拉列表选项，代码如下：

```
<select id="select1">
        <option value="" selected>请选择</option>
        <option value="1">吉林</option>
        <option value="2">辽宁</option>
</select>
<span id='two'></span>
<span id='three'></span>
```

（2）编写 JavaScript 脚本文件，当触发 change 事件时，实现联动效果，代码如下：

```
$("#select1").change( function() {
        var id = $("#select1").val();
        if(id == 1){
                $.get('index.php', null, function(data){
                        $("#two").empty();
                        $("#two").append("<select id='select2'><option>请选择</opton><option value='1'>长春</opton><option value='2'>松原
</opton></select>");
                        $("#select2").change( function() {
                                var page = $("#select2").val();
                                if(page == 1){
                                        $("#three").empty();
                                        $("#three").append("<select><option>双阳</opton><option>农安</opton></select>");
                                }else{
                                        if(page == 2){
                                                $("#three").empty();
                                                $("#three").append("<select><option>乾安</opton><option>大安</opton></select>");
                                        }
                                }
                        });
                });
        }else{
                $.get('index.php', null, function(data){
                        $("#two").empty();
                        $("#two").append("<select id='select3'><option value='1'>大连</opton><option value='2'>旅顺</opton></select>");
                        $("#select3").change( function() {
                                var page = $("#select3").val();
                                if(page == 1){
                                        $("#three").empty();
```

```
                    $("#three").append("<select><option>西岗</opton><option>沙河口</opton></select>");
                }else{
                    if(page == 2){
                        $("#three").empty();
                        $("#three").append("<select><option>金州</opton><option>瓦房店</opton></select>");
                    }
                }
            });
        });
    }
});
```

秘笈心法

心法领悟 163：打造优秀的 3 级联动菜单。

本实例意在讲述实现 3 级联动菜单的原理，在实际开发中是不建议使用 if…else…语句的，因为这样的多层嵌套方式一方面使程序本身的可读性变差，而且还有可能不利于程序员编程。

实例 164　应用下拉列表选择所要联机的网站

光盘位置：光盘\MR\03\164

高级

趣味指数：★★★★

实例说明

在开发网站程序的过程中，多数网站都设置了友情链接这一模块，通常情况下，链接的网站以图标、网站名或地址的形式显示在页面中，浏览者可通过单击超链接进入所要访问的网站。为了更好地利用页面，可以通过下拉列表的方法来创建友情链接。运行本实例，当用户选择下拉列表中所要联机的网站后（如图 3.19 所示），单击 GO 按钮可以立即链接到指定的网站。

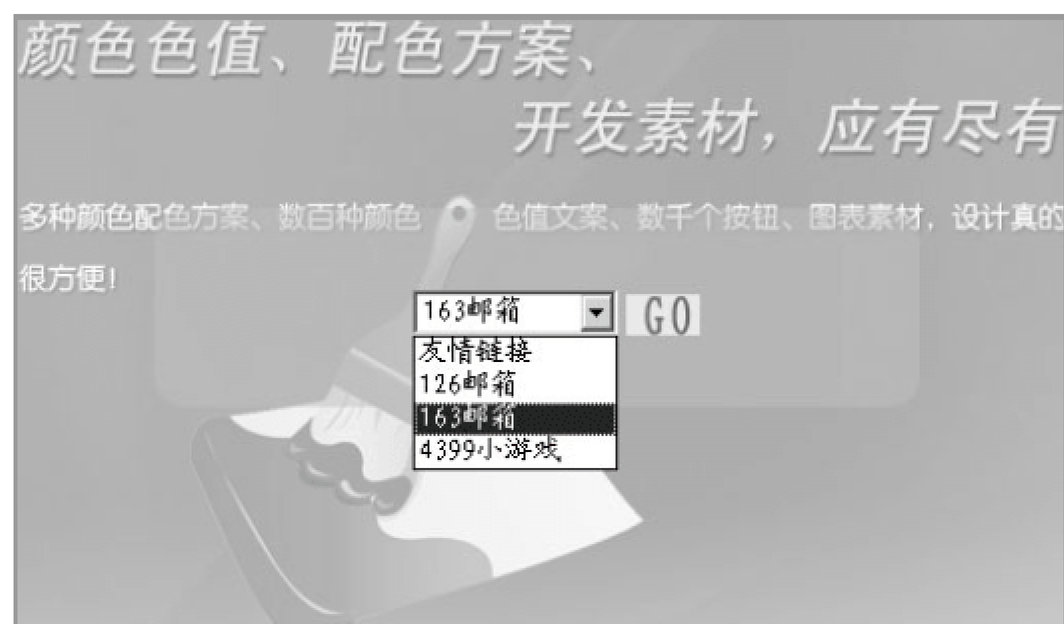


图 3.19　应用下拉列表选择要联机的网站

关键技术

本实例首先将文本框的 value 属性值定义为所要连接的 IP 地址，然后通过设置鼠标的单击事件实现链接到指定的网站。JavaScript 代码如下：

```
$(document).ready(function(){
 $("#button").click(function(){
        var sel = $("#sel").val();
        if(sel != ""){
                $.get(sel);
        }else{
                alert('请选择连接内容');
        }
 });
});
```

设计过程

（1）新建 index.php 文件，创建下拉列表框，并添加下拉列表框选项值，代码如下：

```
<select id="sel">
            <option value="" >友情链接</option>
            <option value="http://www.126.com">126 邮箱</option>
            <option value="http://www.163.com">163 邮箱</option>
            <option value="http://www.4399.com">4399 小游戏</option>
      </select>
      <input id="button" class='button' type="submit" value="       " />
```

（2）编写 JavaScript 脚本文件，当单击 GO 按钮时触发事件，完成页面的跳转操作。

秘笈心法

心法领悟 164：使用 CSS 样式表的好处。

利用 CSS 样式可以大量减少网页代码，其原理就是在网页中自定义样式表的选择符，在网页中大量引用这些选择符即可。

实例 165	可输入字符的下拉菜单 光盘位置：光盘\MR\03\165	高级 趣味指数：★★★★

实例说明

通常，网页中应用的下拉列表的选项值是固定不变的，这样就存在一些弊病，如不能向下拉列表中添加新的选项值。如果能编写可以输入字符的下拉列表，就方便很多。运行本实例，在部门的下拉列表框中有 “生产部”、“业务部”和“技术部”3 条记录，如果要再添加一个“文化部”，直接在下拉列表框中输入“文化部”即可，运行结果如图 3.20 所示。

图 3.20　可输入字符的下拉菜单

关键技术

本实例的开发思路如下：网页中下拉列表的选项值是固定的，要实现在下拉列表中输入文字，首先要捕获用户的键盘事件，然后根据键盘输入的数据动态添加下拉列表的选项，同时还要判断用户按下的是“退格”键还是“空格”键，如果是“退格”键，则需要从光标指针位置向前删除一个字符，如果是“空格”键，则需要在光标指针位置添加一个空格字符。

另外，本实例还利用 onKeyPress 和 onKeyUp 事件调用 JavaScript 自定义函数 press(opt)和 up(opt)。

设计过程

（1）在页面中创建部门下拉列表，其中包括“生产部”、“业务部”和“技术部”3 个选项，代码如下：

```
<form name="form1" method="post" action="index.php" onKeyPress="press(3)" onKeyUp="up(3)">
        <select name="userclass" size="1" id="userclass">
          <option value="生产部" selected>生产部</option>
          <option value="业务部">业务部</option>
          <option value="技术部">技术部</option>
        </select>
  </form>
```

（2）编写键盘按下事件所执行的 JavaScript 自定义函数 press(opt)，用于在下拉列表中添加新的选项，并将用户输入的信息添加到下拉列表中，设置该选项为选中项。

然后编写键盘按下事件所执行的 JavaScript 自定义函数 up(opt)，当用户按 BackSpace 键时，将已经写到下拉列表中的文字逐个删除；当用户按 Space 键时，在下拉列表中添加一个空格字符，代码如下：

```
function press(opt){
//opt 表示现有可选项的数目
 form1.userclass.options[opt]=new Option(ok=(form1.userclass.options[opt])?
 form1.userclass.options[opt].innerText+String.fromCharCode(event.keyCode):
 String.fromCharCode(event.keyCode),ok)
 form1.userclass.selectedIndex=opt;
}
function up(opt){
//opt 表示现有可选项的数目
 if(form1.userclass.options[opt]){
      if(event.keyCode==8){
           var str=form1.userclass.options[opt].innerText;
           var len=str.length;
           form1.userclass.options[opt].innerText=str.substring(0,len-1);
           if(form1.userclass.options[opt].innerText==" ")userclass.remove(2);
      }
      if(event.keyCode==32){
           form1.userclass.options[opt].innerText+=" ";
      }
 }
}
```

秘笈心法

心法领悟 165：使用 JavaScript 代码删除输入文字的注意事项。

在删除输入选项中的文字时，如果删除到最后一个字符，则将该选项一同删除，同时改变下拉列表的选中项。

实例 166	设置下拉列表的默认值 光盘位置：光盘\MR\03\166	高级 趣味指数：★★★★

实例说明

在注册登录模块中，下拉列表往往被应用在选择用户性别上，如果用户忘记选择此下拉列表的内容，就很容易把注册信息弄错。解决这个问题的方法是为下拉列表设置默认值，如图 3.21 所示，当没有对下拉列表进行操作时，将弹出提示信息。



图 3.21　设置下拉列表的默认值

关键技术

本实例首先为下拉列表框设置默认值，然后通过设置按钮的单击事件实现当用户单击提交按钮时，判断下拉列表框的内容是否被选择。

使用 selected 属性实现设置默认值的代码如下：

```
<option selected   value="1">请选择性别</option>
```

设计过程

（1）新建 index.php 文件，引入 CSS 文件和 JavaScript 脚本，编写下拉列表，代码如下：

```
<select>
  <option selected   value="1">请选择性别</option>
      <option >男性</option>
      <option >女性</option>
</select>
<input id="one" type="button" value="        " />
```

（2）编写 JavaScript 代码，当按钮被单击时触发事件，其代码如下：

```
$(document).ready(function(){
 $("#one").click(function(){
         var value = $("select").val();
         if(value == 1){
                  alert("您没有选择性别");
         }
 });
});
```

秘笈心法

心法领悟 166：关于设置下拉列表的默认值。

在使用 selected 时，只需在<option>标记中加入属性名称即可，其代码如下：

```
<option selected="selected"></option>
```

实例 167	设置下拉列表的样式 光盘位置：光盘\MR\03\167	高级 趣味指数：★★★★

实例说明

其实对页面的美化处理，并不属于程序员的工作范畴，而是属于前台美工的份内之事，但是像这种美化下拉列表的小事情，程序员如果还要和美工去协调，未免有点浪费时间，所以用户学习一些基本的页面美化方法

还是很有必要的。运行本实例，效果如图 3.22 所示。



图 3.22　设置下拉列表的样式

关键技术

本实例运用 CSS 样式表改变下拉列表的样式，代码如下：

```
select{
 width:100px;
 font-family:"楷体_GB2312";
 font-size:14px;
 color:#FFFFFF;
 background-color:#000000;
}
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式，编写下拉列表，代码如下：

```
<select>
 <option selected    value="1">请选择性别</option>
   <option >男性</option>
     <option >女性</option>
</select>
```

（2）定义 CSS 样式，代码如关键技术中所示。

秘笈心法

心法领悟 167：关于设置下拉列表的样式。

支持下拉列表的 CSS 样式很少，所以想设置更多的样式，相对来说会很麻烦。如果非要定义一个漂亮的下拉列表样式，最好利用 JavaScript 代码自行编写。

实例 168	下拉列表打开窗口 光盘位置：光盘\MR\03\168	高级 趣味指数：★★★★

实例说明

实例 164 中已经讲解了如何跳转到链接网站的方法，本实例与实例 164 在实现思路上是相同的，唯一不同的是在下拉列表里通过打开新窗口实现跳转。运行本实例，如图 3.23 所示。



图 3.23　选择链接网站

关键技术

本实例的方法是定义 JavaScript 脚本的 change 事件，通过 JavaScript 的 window.open()方法，实现新页面的跳转，代码如下：

```
$(document).ready(function(){
 $("#sel").change(function(){
        window.open($("#sel").val());
 });
});
```

设计过程

（1）创建脚本文件，建立网页框架，引入 CSS 样式表和 JavaScript 脚本，编写下拉列表，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/inc.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<select id="sel" >
        <option>友情链接</option>
        <option value='http://www.126.com'>126 邮箱</option>
        <option value='http://www.163.com'>163 邮箱</option>
        <option value='http://www.4399.com'>4399 小游戏</option>
</select>
```

（2）编写 JavaScript 脚本文件，当发生 change 事件时，发生跳转。

秘笈心法

心法领悟 168：关于 JavaScript。

本章内容主要采用 JavaScript 语言和 jQuery 框架编写，jQuery 框架是以 JavaScript 语言编写的，所以两者可以混编。

实例 169	Tab 键在文本域中的体现 光盘位置：光盘\MR\03\169	高级 趣味指数：★★★★

实例说明

Tab 键在编写 Word 等文档时是经常被用到的。在网页的文本域中，按 Tab 键是进行下一个切换项的指示，如何在文本域中体现 Tab 键呢？本实例将向用户讲解。运行本实例，如图 3.24 所示。

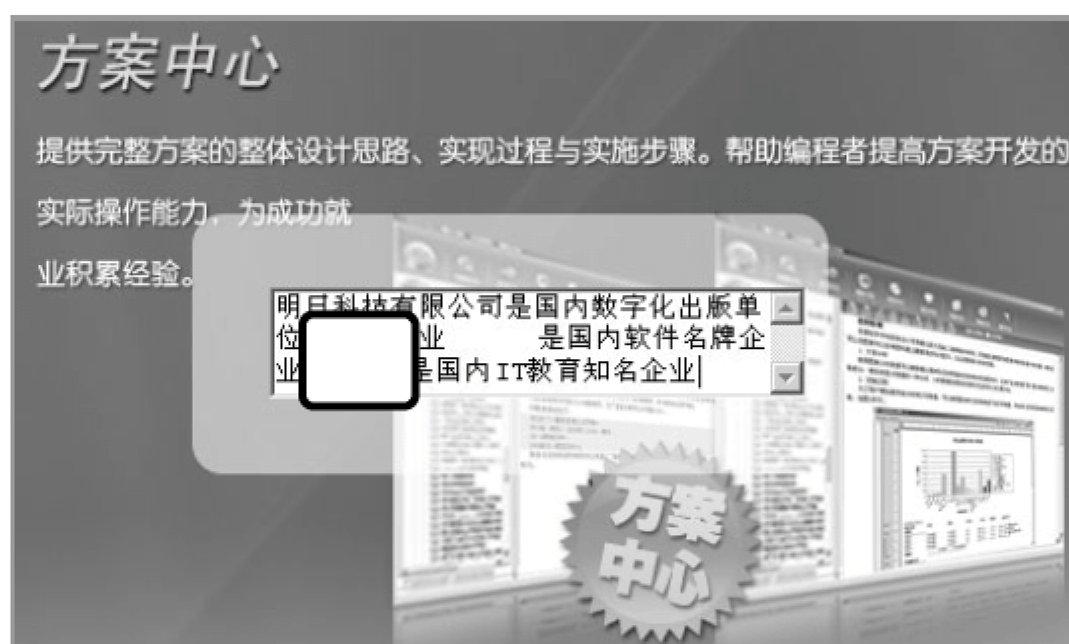


图 3.24　文本域中体现 Tab 键

关键技术

本实例主要是通过 JavaScript 代码定义 onkeydown 事件实现的，代码如下：

```
function editTab(){
    var code, sel, tmp, r
    var tabs=""
    event.returnValue = false
    sel =event.srcElement.document.selection.createRange()
    r = event.srcElement.createTextRange()
    switch (event.keyCode){
        case (8)     :
            if (!(sel.getClientRects().length > 1)){
                event.returnValue = true
                return
            }
            code = sel.text
            tmp = sel.duplicate()
            tmp.moveToPoint(r.getBoundingClientRect().left, sel.getClientRects()[0].top)
            sel.setEndPoint("startToStart", tmp)
            sel.text = sel.text.replace(/^\t/gm, "")
            code = code.replace(/^\t/gm, "").replace(/\r\n/g, "\r")
            r.findText(code)
            r.select()
            break
        case (9)     :
            if (sel.getClientRects().length > 1){
                code = sel.text
                tmp = sel.duplicate()
                tmp.moveToPoint(r.getBoundingClientRect().left, sel.getClientRects()[0].top)
                sel.setEndPoint("startToStart", tmp)
                sel.text = "\t"+sel.text.replace(/\r\n/g, "\r\t")
                code = code.replace(/\r\n/g, "\r\t")
                r.findText(code)
                r.select()
            } else {
                sel.text = "\t"
                sel.select()
            }
            break
        case (13)    :
            tmp = sel.duplicate()
            tmp.moveToPoint(r.getBoundingClientRect().left, sel.getClientRects()[0].top)
            tmp.setEndPoint("endToEnd", sel)
            for (var i=0; tmp.text.match(/^[\t]+/g) && i<tmp.text.match(/^[\t]+/g)[0].length; i++)     tabs += "\t"
            sel.text = "\r\n"+tabs
            sel.select()
            break
        default          :
```

```
                event.returnValue = true
                break
        }
}
```

设计过程

（1）创建脚本文件，引入 JavaScript 脚本，编写文本域标签，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/inc.css"/>
<script type="text/javascript" src="js/in.js"></script>
<textarea cols="30" name="text" rows="3" onkeydown="editTab()"></textarea>
```

（2）当按下键盘按键时触发事件，代码如关键技术中所示。

秘笈心法

心法领悟 169：JavaScript 脚本的引用。

本实例中调用 JavaScript 中的 editTab()完成对 Tab 键的控制，在页面中引用 JavaScript 文件的方法是应用 script 标签，其代码如下：

```
<script type="text/javascript" src="js/in.js"></script>
```

3.4　表单元素的动态操作

所谓动态操作表单元素是指页面中的表单元素并不是固定不变的，而是在程序执行时由系统根据实际情况自动生成。这里将通过实际的例子来介绍如何动态生成表单元素。

实例 170	投票系统中单选按钮与复选框的应用 光盘位置：光盘\MR\03\170	高级 趣味指数：★★★★

实例说明

在开发程序的过程中，经常应用到在线投票模块，其使用非常简单，只要选中一个单选按钮或者多个复选框所对应的投票主题后，单击“提交”按钮即可完成投票。本实例将介绍投票系统的操作原理，当选中一个投票主题之后，就会弹出相应的投票选区，在该选区中可以添加投票，并可以看到投票的结果，运行结果如图 3.25 所示。



图 3.25　在线统计

关键技术

本实例主要是通过文件操作和正则表达式函数实现在线统计程序。其中文件操作的核心代码如下：

```
$file = file_get_contents('info.txt');
        if($file == ""){
                $openn = fopen("info.txt",'a');
                fwrite($openn,'a');
                fclose($openn);
        }
        if(preg_match("/$ip/",$file)){
                echo "<script>alert('同一个 IP 地址，不能投票两次');</script>";
                exit;
        }else{
                if(!$open = fopen("info.txt",'a')){
                        print "文件不存在，请创建";
                        exit;
                }else{
                        fwrite($open,$ip);
                        fclose($open);
                }
        }
```

设计过程

（1）创建脚本文件，利用<table>标签建立网页框架，引入 CSS 文件和 JavaScript 脚本，编写单选按钮，代码如下：

```
<form action="" method="post">
        <tr>
                <td class="three"><input id="id" name="check" checked type="radio" value="1" />上网</td>
                <td><img src="pic/logo1.JPG" /><font color='red' size="-1">10 人</font></td>
        </tr>
        <tr>
                 <td class="three"><input id="id"name="check" type="radio" value="2" />去娱乐场所</td>
                 <td><img src="pic/logo2.JPG" /><font color='red'size="-1">15 人</font></td>
        </tr>
        <tr>
                <td class="three"><input id="id"name="check" type="radio" value="3" />带着女友逛街</td>
                <td><img src="pic/logo3.JPG" /><font color='red'size="-1">20 人</font></td>
        </tr>
        <tr>
                <td class="three"><input id="id"name="check" type="radio" value="4" />睡觉</td>
                <td><img src="pic/logo4.JPG" /><font color='red'size="-1">5 人</font></td>
        </tr>
</form>
```

（2）当单击“确定”按钮时，首先获取客户端 IP 地址，然后将数据写入到文本文件中，最后根据提交的值通过 switch()语句进行判断给出提示信息，代码如下：

```
<?php
 if($_POST[sub]){
        $ip = $_SERVER[REMOTE_ADDR];                          //取得客户端 IP 地址
        $file = file_get_contents('info.txt');                //将文件内容读取到字符串
        if($file == ""){                                      //如果文件内容为空
                $openn = fopen("info.txt",'a');               //打开 info.txt 文件
                fwrite($openn,'a');                           //向文件中写入一个字符
                fclose($openn);                               //关闭文件
        }
        if(preg_match("/$ip/",$file)){                        //匹配字符串、
                echo "<script>alert('同一个 IP 地址，不能投票两次');</script>";
                exit;                                         //退出
        }else{
                if(!$open = fopen("info.txt",'a')){
                        print "文件不存在，请创建";
```

```
                exit;
            }else{
                fwrite($open,$ip);
                fclose($open);
            }
        }
        switch($_POST[check]){                              //switch()选择语句
            case "2";
            echo "<script>alert('您选择去娱乐场所');</script>";
            break;
            case "3";
            echo "<script>alert('您选择和女友逛街');</script>";
            break;
            case "4";
            echo "<script>alert('您选择睡觉');</script>";
            break;
            case "1";
            echo "<script>alert('您选择上网');</script>";
            break;
        }
}
?>
```

秘笈心法

心法领悟 170：关于对文件的操作。

文件的操作，对于所有学习 PHP 这门语言的人来说，都会感到头疼，特别是对文件的高级操作中的指针的理解，更是如此。所以笔者建议，在学习这部分知识时，一定要先打好基础，然后多实践、多思考。

实例 171　选中单选按钮后显示其他表单元素

光盘位置：光盘\MR\03\171

高级

趣味指数：★★★★

实例说明

论坛系统中的会员有不同的级别，一种是超级会员，拥有对论坛的管理权限；另一种是普通会员，只能发布和回复论坛信息。因此在论坛的权限设置模块中就需要应用单选按钮对超级会员和普通会员设置不同的权限。在本实例中，选中“超级会员”单选按钮时，其下方将列出网站的所有管理模块，管理员可以为超级会员设置管理权限，运行结果如图 3.26 所示；选中“普通会员”单选按钮，运行结果如图 3.27 所示。



图 3.26　超级会员设置

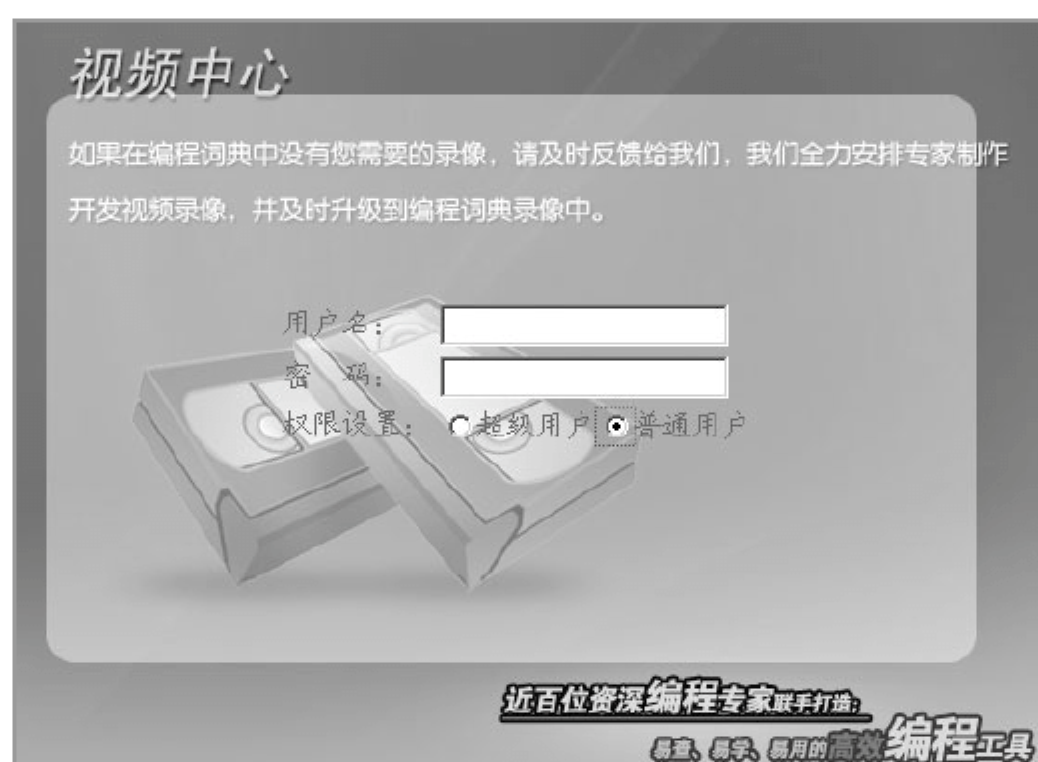


图 3.27　普通会员设置

关键技术

选中单选按钮后显示其他表单元素的功能，主要利用 JavaScript 设置表格的 display 样式，来实现表格的显示和隐藏。通过设置表格的 ID 属性，可以实现对表格的整体控制，如设置表格的 style（样式）属性。通过将表格的 display 样式设置为“none”，可以控制表格的隐藏；将表格的 display 样式设置为“block”，可以显示表格。

设计过程

（1）创建脚本文件，利用<table>标签编写网页架构，引入 CSS 样式表和 JavaScript 脚本，编写<form>表单，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table align="center">
  <tr>
    <td>用户名：</td>
    <td><input type="text"/></td>
  </tr>
  <tr>
    <td>密  码：</td>
    <td><input type="text"/></td>
  </tr>
  <tr>
    <td>权限设置：</td>
    <td><input id="va" type="radio" name="ch" value="1" checked/>超级用户<input id="val" name="ch"type="radio" value="2" />普通用户</td>
  </tr>
  <tr>
        <td><table id="name1"><tr><td><input type="checkbox" />帖子管理</td></tr></table></td>
        <td><table id="name2"><tr><td><tr><td><input type="checkbox" />后台管理</td></tr></table></td>
  </tr>
  <tr>
        <td><table id="name3"><tr><td><tr><td><input type="checkbox" />数库管理</td></tr></table></td>
        <td><table id="name4"><tr><td><tr><td><input type="checkbox" />公告管理</td></tr></table></td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件，将<style>属性动态添加到复选框的表格中，控制表格的隐藏和显示，代码如下：

```
$(document).ready(function(){
 $("#va").click(function(){
        $("#name1").css("display","block");
        $("#name2").css("display","block");
        $("#name3").css("display","block");
        $("#name4").css("display","block");
 });
 $("#val").click(function(){
        $("#name1").css("display","none");
        $("#name2").css("display","none");
        $("#name3").css("display","none");
        $("#name4").css("display","none");
 });
});
```

秘笈心法

心法领悟 171：添加 CSS 属性。

利用 jQuery 框架可以很容易地将属性和值动态地添加到指定标签中；语法格式为：

```
$("class/id").attr("key","value");
```



实例 172　不提交表单获取单选按钮的值

光盘位置：光盘\MR\03\172

高级

趣味指数：★★★★

实例说明

PHP 这门语言是一种服务器端的编程语言，顾名思义，在需要时与服务器进行交互，并且处理信息，但是很多时候，用户希望在与服务器交互之前就完成一些操作。这个时候客户端脚本语言 JavaScript 的作用就体现出来了。运行本实例，通过 JavaScript 代码实现不提交表单获取单选按钮的值，如图 3.28 所示。

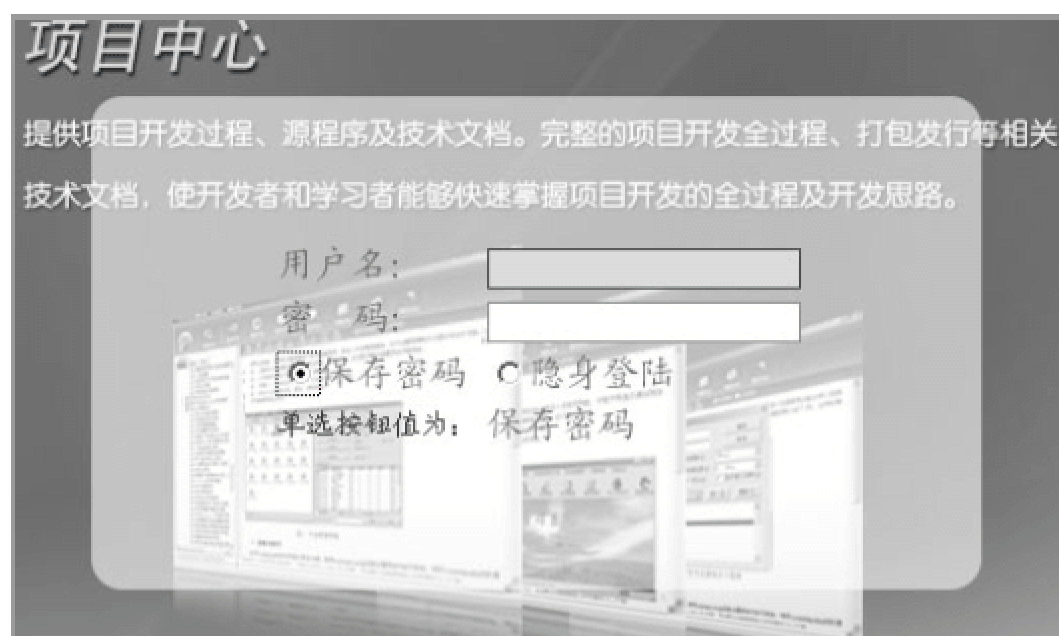


图 3.28　不提交表单获取单选按钮的值

关键技术

本实例主要是通过 JavaScript 代码定义 click 事件，获取单选按钮的值，并且将单选按钮的值通过 div 标签返回到页面中，代码如下：

```
$(document).ready(function(){
 $(".n").mouseover(function(){
        $(".n").css({background:"#FFCCCC",border:"1px red solid"});
        $(".n").mouseout(function(){
                $(".n").css({background:"white",border:"1px #00CCFF solid"});
        });
 });
 $(".o").mouseover(function(){
        $(".o").css({background:"#FFCCCC",border:"1px red solid"});
        $(".o").mouseout(function(){
                $(".o").css({background:"white",border:"1px #00CCFF solid"});
        });
 });
 $("#e").click(function(){
        $("#nam").css("display","block");
        var value = $("#e").val();

        $("#name").html(value);
 });
 $("#ee").click(function(){
        $("#nam").css("display","block");
        var value = $("#ee").val();

        $("#name").html(value);
 });
});
```

设计过程

（1）创建脚本文件，引入 CSS 样式和 JavaScript 脚本文件，创建用户登录的 form 表单，设置文本框和单

选按钮元素，并且设置 div 标签，定义标签的 ID 值，核心代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table align="center">
  <tr>
    <td>用户名：</td>
    <td><input class="n" type="text" /></td>
  </tr>
  <tr>
    <td>密  码：</td>
    <td><input class="o" type="text" /></td>
  </tr>
  <tr>
    <td><input id="e" name="ch"type="radio" value="保存密码" />保存密码</td>
    <td><input id="ee" name="ch"type="radio" value="隐身登录" />隐身登录</td>
  </tr>
   <tr>
    <td id="nam">单选按钮值为：</td>
    <td><div id="name"></div></td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，引入 jQuery 库文件 jquery-1.3.2.js。

秘笈心法

心法领悟 172：jQuery 中的 click 事件。

当鼠标单击标签时触发此事件，语法格式如下：

```
$("class/id").click(function(){……});
```

实例 173 实现复选框中的全选、反选和不选

光盘位置：光盘\MR\03\173

高级

趣味指数：★★★★

实例说明

复选框应用在用户注册系统中，例如个人爱好选择等。本实例所讲述的 3 个操作：全选、反选、不选，只是根据实际情况，对复选框实现的快捷操作。运行本实例，全选的效果如图 3.29 所示。

图 3.29 实现全选

关键技术

复选框的全选、反选和不选主要应用 JavaScript 脚本，通过自定义函数完成。

（1）在通过 JavaScript 脚本中的自定义函数完成复选框的全选、反选和不选功能中，应用的第 1 个方法是 getElementsByTagName，获取指定标签的名称，返回值为一个包含标签信息的 object。

（2）根据 getElementsByTagName 标签返回的对象，判断标签类型（type）的值是否是 checkbox。

（3）当标签类型 type 的值为 checkbox 时，为标签中的 checked 赋值。当 checked 的值为 TRUE 时，将其赋值为 FALSE；当 checked 的值为 FALSE 时，将其赋值为 TRUE。

（4）在页面中通过 script 标签调用 js 文件夹中的文件，应用 onClick 事件调用自定义函数，完成全选、反选和不选功能的实现。

设计过程

（1）创建 js 文件夹，编写 reg.js 脚本文件。在 reg.js 中，编写自定义函数，实现全选、反选和不选功能。reg.js 文件的关键代码如下：

```
function uncheckAll(form1,status) {                                //不选
 var elements = form1.getElementsByTagName('input');               //获取 input 标签
 for(var i=0; i<elements.length; i++){                             //根据标签的长度执行循环
        if(elements[i].type == 'checkbox') {                       //判断对象中元素的类型
                if(elements[i].checked==true){                     //判断当 checked 的值为 TRUE 时
                        elements[i].checked=false;                 //将 checked 赋值为 FALSE
                }
        }
 }
}
function checkAll(form1,status){                                   //全选
 var elements = form1.getElementsByTagName('input');
 for(var i=0; i<elements.length; i++){
        if(elements[i].type == 'checkbox') {
                if(elements[i].checked==false){
                        elements[i].checked=true;
                }
        }
 }
}
function switchAll(form1,status) {                                 //反选
 var elements = form1.getElementsByTagName('input');
 for(var i=0; i<elements.length; i++){
        if(elements[i].type == 'checkbox'){
                if(elements[i].checked==true){
                        elements[i].checked=false;
                }else if(elements[i].checked==false){
                        elements[i].checked=true;
                }
        }
 }
}
```

（2）创建 index.php 页面，输出会员信息，并添加图像按钮，通过 onClick 事件调用 JavaScript 自定义函数实现全选、反选、不选和删除的功能，其关键代码如下：

```
<script language="javascript" src="js/reg.js"></script>
<form method="post" name="form1" id="form1" action="index_ok.php">
<tr>
      <td width="62" align="center"><input name="conn_id[]" type="checkbox" id="conn_id[]" value="1" /></td>
      <td width="242">欧阳</td>
      <td width="243">PHP</td>
      <td width="485">部门经理</td>
      <td width="485">29</td>
```

```
</tr>
<tr>
<td colspan="5" align="center"><!--通过 onClick 事件调用自定义函数，执行相应的操作-->
<img src="images/bg_19-20.jpg" onclick="checkAll(form1,status)" width="62" height="25" />
      <img src="images/bg_14-14.jpg" onclick="switchAll(form1,status)"   width="62" height="25" />
      <img src="images/bg_07-08.jpg" width="62" height="25" onclick="uncheckAll(form1,status)" />
<input type="image" name="imageField" src="images/bg_14.jpg" />
</td>
</tr>
</form>
```

秘笈心法

心法领悟 173：复选框的全选、反选和不选的实现思路。

复选框的全选、反选和不选主要应用 JavaScript 脚本，通过 JavaScript 脚本中的自定义函数，判断复选框中 checked 的值。全选就是将 checked 赋值为 TRUE，不选则是将 checked 赋值为 FALSE，而反选则是当 checked 的值为 TRUE 时，将其赋值为 FALSE；当 checked 的值为 FALSE 时，将其赋值为 TRUE。

实例 174 隐藏域提交用户的 ID 值

光盘位置：光盘\MR\03\174

高级

趣味指数：★★★★

实例说明

开发在线论坛系统时，在回复帖子的模块中，要记录是谁回复了哪一个帖子，即要把这个回复用户的 ID 同回复帖子的内容一起保存起来。回帖的内容是从文本域中获取的，而回帖人的 ID 则是通过隐藏域进行提交的，不需要手动填写，如图 3.30 所示。

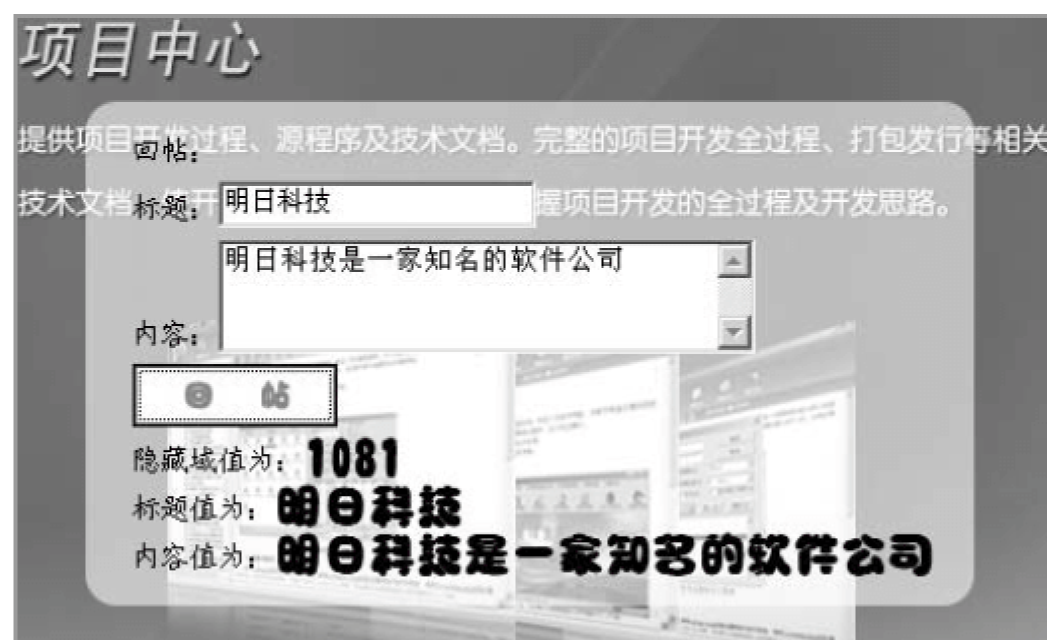


图 3.30　隐藏域提交用户 ID

关键技术

本实例主要应用表单中的隐藏域来实现文件值的提交和获取。

表单中隐藏域的语法格式如下：

```
<input type= "hidden" name="ID" value="">
```

参数说明：

type：表单的类型，其中的 hidden 表示隐藏域。

name：隐藏域的名称，可以自己定义。

value：隐藏域的值，可以填写隐藏域的默认值。

设计过程

（1）创建脚本文件，利用<table>标签建立网页框架，引入 CSS 样式和 JavaScript 脚本，编写表单，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table width="600px" height="600px" background="pic/bg.jpg">
  <tr>
    <td>
      <table align="center">
  <tr>
    <td>回帖：<input type="hidden" name="ID" value="1081" /></td>
  </tr>
  <tr>
    <td>标题：<input type="text"   /></td>
  </tr>
  <tr>
    <td>内容：<textarea cols="30" rows="3"></textarea></td>
  </tr>
  <tr>
    <td><input class="one" type="submit" name="sub" value="" /></td>
  </tr>
  <tr><td id="name" style="display:none">
  隐藏域值为：<b id="b1"></b><br>标题值为：<b id="b2"></b><br>内容值为：<b id="b3"></b>
  </td></tr>
</table>
</td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本，当单击“回帖”按钮时触发事件，代码如下：

```
$(document).ready(function(){
$(".one").click(function(){
        var val1 = $("input:eq(0)").val();
        var val2 = $("input:eq(1)").val();
        var val3 = $("textarea").val();
        $("#name").css("display","block");
        $("#b1").html(val1);
        $("#b2").html(val2);
        $("#b3").html(val3);
});
});
```

秘笈心法

心法领悟 174：jQuery 中使用$(“”).html()方法的注意事项。

此方法是取得第 1 个匹配元素的 html 内容。这个函数不能用于 XML 文档，但可以用于 XHTML 文档。

实例 175	图像域替代提交按钮 光盘位置：光盘\MR\03\175	高级 趣味指数：★★★★

实例说明

在很多网站的主页布局中，程序员除了在导航栏中布置一些按钮外，在网页的主体部分中很少出现类似于提交的按钮，这样做的主要目的并不是因为网页中不再需要按钮，而是被一些超链接或者图像域跳转所替代。运行本实例，当单击图片时实现表单提交，如图 3.31 所示。

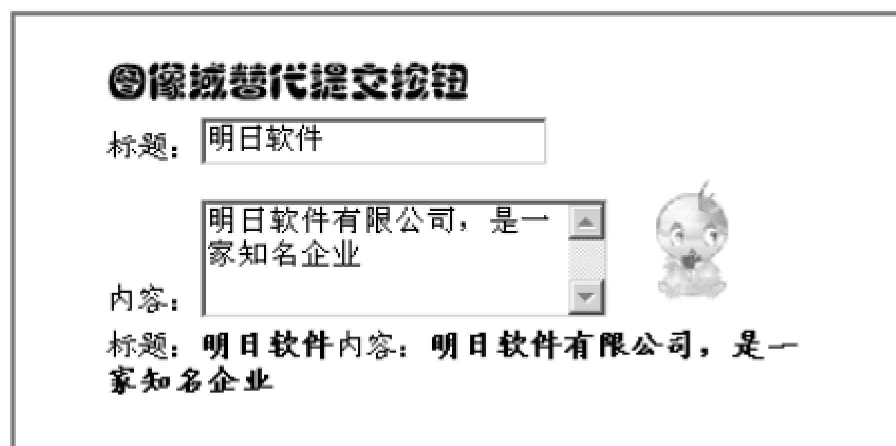


图 3.31　图像域代替提交按钮

关键技术

本实例主要是利用 JavaScript 代码，通过单击事件完成数据的提交操作，其关键代码如下：

```
$(document).ready(function(){
 $("img").click(function(){
        var input = $("input").val();
        var textarea = $("textarea").val();
        if(input == "" || textarea == ""){
                alert("文本框为空");
        }else{
                $(".n").css("display","block");
                $("#i").html(input);
                $("#t").html(textarea);
        }
 });
});
```

设计过程

（1）创建脚本文件，建立网页框架，引入 CSS 样式和 JavaScript 脚本，编写表单，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table width="319" align="center">
  <tr>
    <td width="296" class="oo">图像域替代提交按钮</td>
  </tr>
  <tr>
    <td>标题：<input type="text" /></td>
  </tr>
  <tr>
    <td>内容：<textarea cols="20" rows="3"></textarea>  <img src="pic/img.gif" /></td>
  </tr>
  <tr>
    <td class="n">标题：<b id="i"></b>内容：<b id="t"></b></td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，当单击图标时触发事件，代码如关键技术所示。

秘笈心法

心法领悟 175：图像域的使用。

使用图像域进行跳转和提交的方式有很多。例如嵌入到<a>标签中，进行客户端跳转，或者应用在<form>表单的提交、重置按钮中。灵活运用图像域的知识会使您的程序更美观。

实例 176　跳转菜单实现页面跳转

光盘位置：光盘\MR\03\176

高级

趣味指数：★★★★

实例说明

跳转菜单在网站中是很常见的，一般被放置在靠近头部的位置，又称导航栏。下面就向用户讲解如何实现单击按钮页面跳转的效果，如图 3.32 所示。



图 3.32　跳转菜单实现页面跳转

关键技术

本实例主要通过<map>定义跳转，本实例实现跳转的代码如下：

```
<map name="Map2" id="Map2">
  <area shape="rect" coords="89,64,141,94" href="index.php?id=1" />
  <area shape="rect" coords="168,64,229,94" href="index.php?id=2" />
  <area shape="rect" coords="250,65,305,99" href="index.php?id=3" />
  <area shape="rect" coords="328,63,388,97" href="index.php?id=4" />
  <area shape="rect" coords="13,65,61,93" href="index.php" />
</map>
```

设计过程

（1）创建 index.php 文件，设计网页，通过<map>定义页面跳转的超链接。

（2）编写 PHP 代码，当单击不同按钮时实现跳转，代码如下：

```
<?php
      switch($_GET['id']){
            case "1";
            echo "pic/reg.JPG";
            break;
            case "2";
            echo "pic/log.JPG";
            break;
```

```
            case "3";
            echo "pic/for.JPG";
            break;
            case "4";
            echo "pic/mes.JPG";
            break;
            default:
            echo "pic/for.JPG";
            break;
        }
?>
```

秘笈心法

心法领悟 176：如何取得地址栏参数。

以本实例为例子，地址组成格式如下所示，可以通过$_GET 方式取得相应的参数并进行操作。

```
http://地址/主页.php?参数&参数
```

实例 177　上传图片预览

光盘位置：光盘\MR\03\177

高级

趣味指数：★★★★

实例说明

上传图片时最好为程序设置预览功能，防止用户意外传错图片。还有一个很重要的模块就是获取图片的地址，方便用户网站的推广。运行本实例，完成预览图片和获取图片地址的功能，运行结果如图 3.33 所示。



图 3.33　上传预览图片

关键技术

本实例通过 JavaScript 的 click 事件，实现图片预览、获取地址和清除上传文本框信息的功能，其核心代码如下：

```
$(document).ready(function(){
 $("#two").click(function(){                                    //单击预览图片
        var value = $("#one").val();                            //取得上传地址
        if(value == ""){                                        //判断地址是否为空
            alert("地址为空");
        }else{
            $(".n").append("<img id='five' src="+value+">");    //显示图片
            $("#two").click(function(){
                $(".n").empty();
                $(".n").append("<img id='five' src="+value+">");
```

```
                    });
            }
});
$("#three").click(function(){                                        //显示图片地址
        var value = $("#one").val();
        if(value == ""){
                alert("地址为空");
        }else{
                alert(value);
        }
});
$("#four").click(function(){                                         //清除文本框信息
        $(".e").empty();
        $(".e").append("<input class='one' id='one' type='file'>");
});
});
```

设计过程

（1）新建 index.php 文件，引入 CSS 样式、in.js 文件以及 jQuery 库文件 jquery-1.3.2.js。创建 form 表单，设置文件域属性，添加预览图片、图片地址和清除的按钮，代码如下：

```
<table width="600" height="450" align="center" background="pic/bg.jpg">
   <tr>
      <td align="center"><div class="n"></div>
<table align="center">
   <tr>
<td>
        <span class="o">
        <span class="e"><input class="one" id="one" type="file"></span>
        <input class="one" id="two" type=button value="预览图片">
        <input class="one" id="three" type=button value="图片地址">
        <input class="one" id="four" type=button value="清除" >
        </span>
</td>
   </tr>
</table>
</td>
   </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，代码如关键技术中所示。

秘笈心法

心法领悟 177：jQuery 框架中的 empty()方法。

empty()方法：删除匹配的元素集合中所有的子节点，语法格式如下：

```
$('class/id').empty();
```

参数说明：

class/id：表示类名或 ID 名称。

实例 178　去掉下拉选项的边框

光盘位置：光盘\MR\03\178

高级　趣味指数：★★★★

实例说明

由于 CSS 样式对下拉列表框的定义很少，以至于想要修改下拉列表的样式变得很复杂。本实例将向用户讲解如何去掉下拉选项的边框，如图 3.34 所示。

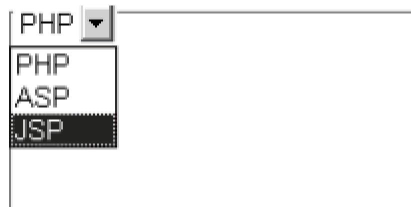


图 3.34　去掉下拉选项的边框

关键技术

本实例的关键点是严格定义 CSS 样式，代码如下：

```
<div style="position: absolute; left: 10px; top: 10px; width: 115px; height: 20px;   clip:rect(2 114 20 2);">
```

设计过程

创建 index.php 文件，利用<table>标签编写表格边框。编辑 CSS 样式，通过 CSS 样式去掉下拉选项的边框，代码如下：

```
<table width="200px" height="100px" border="1" bordercolor="#0099FF" cellspacing="0"><tr><td>
<div style="position: absolute; left: 10px; top: 10px; width: 115px; height: 20px;   clip:rect(2 114 20 2);">
    <select>
      <option >PHP</option>
      <option >ASP</option>
    <option >JSP</option>
   </select>
</div>
</td></tr></table>
```

秘笈心法

心法领悟 178：CSS 的效果。

像这样较为生僻的代码，其实笔者也不能够完全吃透，但是没关系，因为我们所从事的是后台的开发，至于前台的东西，多了解一下即可。

实例 179	修改表单属性为弹出窗口 光盘位置：光盘\MR\03\179	高级 趣味指数：★★★★

实例说明

表单可以添加属性并且与客户端脚本 JavaScript 交互，例如 JavaScript 的 click 事件、change 事件等。运行本实例，当单击文本框时弹出提示信息，如图 3.35 所示。



图 3.35　弹出窗口

关键技术

本实例主要是通过 JavaScript 代码定义单击事件，核心代码如下：

```
$(document).ready(function(){
 $("#o").click(function(){
        alert("文本框被单击");
```

```
});
});
```

设计过程

创建 index.php 文件，引入 CSS 样式、JavaScript 脚本文件 in.js 和 jQuery 库文件 jquery-1.3.2.js。创建 form 表单并定义事件，代码如下：

```
<table align="center">
  <tr>
    <td>用户名：</td>
    <td><input id="o" type="text" /></td>
  </tr>
  <tr>
    <td>密  码：</td>
    <td><input id="o" type="text" onclick="alert('密码框被单击')"/></td>
  </tr>
</table>
```

秘笈心法

心法领悟 179：定义 JavaScript 事件。

如果用户想利用 JavaScript 本身的方法定义事件，需要在表单中进行，代码格式如实例 180 所示。如果是利用 jQuery 框架定义事件，只需要在标签中定义 ID 或者类即可。

```
<input id="o"type="text" onclick="alert('密码框被单击')"/>
```

实例 180	表单输入单元的文字设置 光盘位置：光盘\MR\03\180	高级 趣味指数：★★★★

实例说明

灵巧的 CSS 是所有 Web 开发者追捧的对象，其主要作用是通过简单的代码美化页面，页面美化几乎用 CSS 都可以实现。本实例主要讲解通过定义 CSS 样式对输入单元的文字进行设置，如图 3.36 所示。

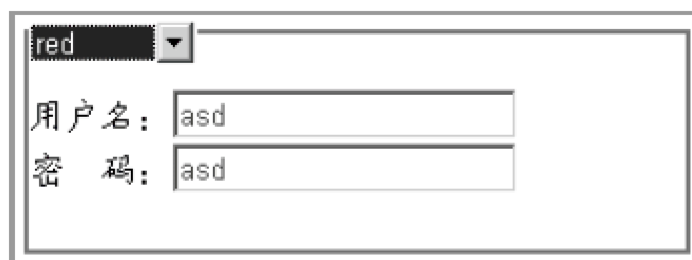


图 3.36　动态改变文字颜色

关键技术

本实例的关键点是定义 CSS 样式，通过 JavaScript 代码动态地改变文本框的颜色。JavaScript 代码如下：

```
<script type="text/javascript" language="javascript">
$(document).ready(function(){
        $("select").change(function(){
                var value = $("select").val();
                if(value == ""){
                        alert("请选择样式颜色");
                }else{
                        $("input:lt(2)").css("color",value);
                }
        });
});
</script>
```

设计过程

（1）创建 index.php 文件，利用 CSS 样式定义一个无边框的下拉列表，代码如下：

```
<div style="position: absolute; left: 10px; top: 10px; width: 115px; height: 20px;    clip:rect(2 114 20 2);">
   <select>
     <option value="">样式颜色</option>
     <option value="red">red</option>
     <option value="blue">blue</option>
     <option value="yellow">yellow</option>
   </select>
</div>
```

（2）编写文本框和密码框，代码如下：

```
<div style="font-family:'楷体_GB2312'; font-size:16px;">
用户名：<input type="text" /><br>
密  码：<input type="text" />
</div>
```

（3）编写 JavaScript 脚本文件 in.js，动态改变文本框输入单元颜色，代码如关键技术中所示。

秘笈心法

心法领悟 180：jQuery 中匹配所有小于给定索引值的元素。

格式，例如$("tr:lt(2)")，匹配标签<tr>中小于索引值 2 的元素。

实例 181	表单输入单元单击删除 光盘位置：光盘\MR\03\181	高级 趣味指数：★★★★

实例说明

在创建表单中的文本框等内容时，往往会给文本框设定初始值，这些初始值需要用户自行手动删除。为了方便用户，节省操作时间，可以通过设置 JavaScript 事件，实现直接删除文本框中初始值的功能。运行本实例，如图 3.37 所示。

图 3.37　表单输入单元单击删除

关键技术

本实例主要是通过 JavaScript 的 click 事件实现，当单击文本框时删除文本框中的默认值，代码如下：

```
$(document).ready(function(){
 $("input").click(function(){
         $(this).attr("value","");
```

```
        alert('清除文本框信息');
});
$("textarea").click(function(){
        $(this).attr("value","");
});
});
```

设计过程

创建 index.php 文件，引入 CSS 样式、JavaScript 脚本文件 in.js 和 jQuery 库文件 jquery-1.3.2.js。编写表单信息，代码如下：

```
<table align="center">
  <tr>
    <td>标题：</td>
    <td><input type="text" value="请输入标题" /></td>
  </tr>
  <tr>
    <td>内容：</td>
    <td><textarea cols="23" rows="3">请输入内容</textarea></td>
  </tr>
</table>
```

秘笈心法

心法领悟 181：如何简单实现本实例功能。

本实例是采用 jQuery 框架实现的，书写代码较多，还有一种简便方法是在表单中直接定义 onFocus 事件，代码如下：

```
<input type='text' name='te' value='请输入内容' onFocus="this.value=''">
```

实例 182　表单文本输入的移动选择

光盘位置：光盘\MR\03\182

高级

趣味指数：★★★★

实例说明

很多网站中，往往有一些小的技巧，例如，在文本域中输入内容时，通过下拉列表动态地将文字插入到文本域中。下面就向用户讲述这一功能。运行本实例如图 3.38 所示。

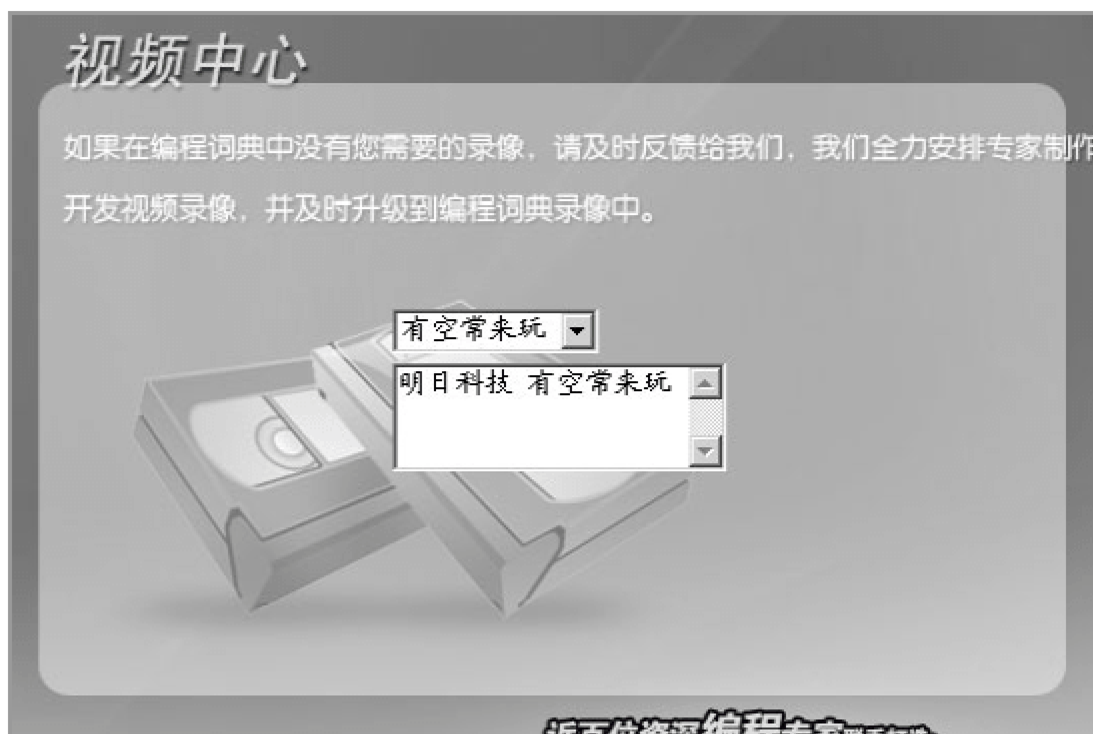


图 3.38　表单文本输入的移动选择

关键技术

本实例通过 JavaScript 脚本中的 change 事件，实现动态插入信息，其核心代码如下：

```
$(document).ready(function(){
 $("textarea").select();                          //选择下拉列表
 $("select").change(function(){                   //定义事件
        var value = $(this).val();                //获取 value 值
        var va = $("textarea").val();             //获取文本域的值
        if(va == "请输入内容"){
                $("textarea").html(value);        //输出文本域的值
        }else{
                $("textarea").html(va+value);     //为文本域赋值
        }
 });
});
```

设计过程

（1）创建 index.php 文件，引入 CSS 样式和 JavaScript 脚本文件 in.js，以及 jQuery 库，创建下拉列表框和文本域，代码如下：

```
<table align="center">
  <tr>
    <td><select name="select">
      <option value="谢谢">谢谢</option>
      <option value="不客气">不客气</option>
      <option value="有空常来玩">有空常来玩</option>
      <option value="再见">再见</option>
</select></td>
  </tr>
  <tr>
    <td><textarea cols="20" rows="3">请输入内容</textarea></td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，代码如关键技术中所示。

秘笈心法

心法领悟 182：如何利用 jQuery 实现文本框内容全选。

实现文本框内容全选与单击文本框清除所有信息所达到的效果大同小异，其实现代码如下：

```
$("textarea").select();
```

实例 183 通过下拉列表选择头像

光盘位置：光盘\MR\03\183

高级

趣味指数：★★★★

实例说明

在腾讯的 QQ 系统中，我们可以通过单击头像实现更换新头像的功能。其实这个功能相对来说是比较简单的。运行本实例，通过下拉列表实现头像的选择，如图 3.39 所示。



图 3.39 选择头像

关键技术

本实例通过 JavaScript 的 change 事件和实例 162 中的 2 级联动菜单实现，其核心代码如下：

```
$(document).ready(function(){
 $("select").change(function(){
        var value = $(this).val();
        if(value == ""){
                alert("请选择性别");
        }else{
                if(value == "nan"){
                        $("#o").empty();
                        $("#o").append("<select id='one'><option value='pic/1.jpg'>头像 1</option><option value='pic/2.jpg'>头像 2</option><option
value='pic/3.jpg'>头像 3</option><option value='pic/4.jpg'>头像 4</option><option value='pic/9.jpg'>头像 5</option></select>");
                        $("#n").empty();
                        $("#n").append("<img src='pic/1.jpg'>");
                        $("#one").change(function(){
                                var va = $(this).val();
                                $("#n").empty();
                                $("#n").append("<img src="+va+">");
                        });
                }else{
                        $("#o").empty();
                        $("#o").append("<select id='one'><option value='pic/5.jpg'>头像 1</option><option value='pic/6.jpg'>头像 2</option><option
value='pic/7.jpg'>头像 3</option><option value='pic/8.jpg'>头像 4</option><option value='pic/10.jpg'>头像 5</option></select>");
                        $("#n").empty();
                        $("#n").append("<img src='pic/5.jpg'>");
                        $("#one").change(function(){
                                var va = $(this).val();
                                $("#n").empty();
                                $("#n").append("<img src="+va+">");
                        });
                }
        }
 });
});
```

设计过程

（1）创建 index.php 文件，引入 CSS 样式和 JavaScript 脚本文件 in.js，以及 jQuery 库，编写下拉列表和标签，代码如下：

```
<table align="center">
  <tr>
    <td>
      <select name="select">
              <option value="">请选择性别</option>
              <option value="nan">男性</option>
              <option value="nv">女性</option>
      </select>
      <div id="o"></div><div id="n"></div>
</td>
  </tr>
</table>
```

（2）编写 JavaScript 脚本文件 in.js，代码如关键技术中所示。

秘笈心法

心法领悟 183：　jQuery 中的 this 关键字。

jQuery 中的 this 关键字与其他编程语言大致相同，此关键字表示当前方法体的类名称或者 ID 和标签，书写格式如下：

```
var va = $(this).val();
```

3.5 CSS+DIV 页面布局

CSS+DIV 是一种网站的标准（或称面向 Web 的标准）中常用术语之一，通常是为了说明与 HTML 页面中的表格（table）定位技术的区别，或者是因为 HTML 网站设计标准中，不能使用表格定位技术，而采用 CSS+DIV 方式实现的各种定位。

实例 184	应用 CSS+DIV 实现无边框窗口 光盘位置：光盘\MR\03\184	高级 趣味指数：★★★★

实例说明

对于一个页面风格清新自然的网站来说，如果打开的窗口带有不适合页面风格的灰色边框及死板的标题栏，不但不能体现其个性化，而且势必会影响网站的整体效果。本实例介绍一种应用 CSS+DIV 实现的无边框窗口。运行本实例，单击“用户登录”超链接，即可打开无边框的用户登录窗口，如图 3.40 所示。



图 3.40 无边框窗口

关键技术

本实例通过 CSS+DIV 排版定位技术，将表格放置在指定位置，CSS 代码如下：

```
body{
margin:0;
text-align:center;
}
#container{
position:relative;
margin:10px 10px 10px 10px;
width:600px;
height:600px;
text-align:center;
background:url(../pic/bgc.jpg);
}
#NOborder{
clear:left;
float:right;
position:absolute;
```

```
margin-right:10px;
width:200px;
height:100px;
margin:auto;
}
#border{
width:auto;
height:100px;
}
#href{
clear:left;
float:right;
position:absolute;;
margin:40px 10px 300px 200px;
width:80px;
height:20px;
}
```

并利用 JavaScript 代码实现表格的显示或隐藏，JavaScript 代码如下：

```
$(document).ready(function(){
$("a").click(function(){
        $("table").css("display","block");
});
$("#two").click(function(){
        $("table").css("display","none");
});
});
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式文件和 JavaScript 脚本文件，通过<div>标签对网页进行布局，编写无边框表格，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<div id="container">
<div id="border">
        <div id="href">
                <a href="#">用户登录</a>
        </div>
</div>
<div id="NOborder">
<table border="1" bordercolor="#999999" cellspacing="0"><tr><td>
<table>
   <tr>
      <td>==用户登录==</td>
   </tr>
   <tr>
      <td>用户名：<input type="text" size="10"/></td>
   </tr>
   <tr>
      <td>密  码：<input type="text" size="10" /></td>
   </tr>
   <tr>
      <td><input id="one" value="" type="button" /><input id="two" value="" type="button" /></td>
   </tr>
</table>
</td></tr></table>
</div>
</div>
```

（2）编写 CSS 样式文件 in.css 和 JavaScript 脚本文件 in.js，代码如关键技术中所示。

秘笈心法

心法领悟 184：CSS 中的 position 关键字的使用。

position：检索对象的定位方式。

参数说明：

static：无特殊定位，对象遵循 HTML 定位规则。

Absolute：将对象从文档流中拖出，使用 left、right、top、bottom 等属性进行绝对定位，而其层叠通过 z-index 属性定义。此时对象不具有边距，但仍有补白和边框。

relative：对象不可层叠，但将依据 left、right、top、bottom 等属性在正常文档流中偏移位置。

fixed：IE 5.5 及 NS 6 尚不支持此属性。

实例 185 | 设置表格样式 | 高级

光盘位置：光盘\MR\03\185

趣味指数：★★★★

实例说明

表格在 Web 开发中非常常见，很多程序员都喜欢利用表格对自己数据库中的信息进行格式输出。但是表格在默认情况下，无论从色调还是格式上都不是很理想，这就需要程序员利用 CSS 的知识对表格样式进行设置。运行本实例，如图 3.41 所示。

用户ID	用户名	密码	操作时间
001	ym	***	2010-08-12
002	pkh	***	2010-08-23
003	lzh	***	2010-07-06

图 3.41　设置表格样式

关键技术

本实例利用 CSS 样式，对表格的边框、背景色等进行设置，核心代码如下：

```
body{
 margin-top:150px;
}
#o{
 margin-left:150px;
}
table,select{
 font-family:"楷体_GB2312";
 font-size:16px;

}
table{
 background:#32ACEF;
}
td{
 text-align:center;
}
.one{
 background:#FFFFFF;
}
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式文件 in.css，利用<div>标签设计网页框架，代码如下：

```
<div id="o">
 <div id="banner">
        <img src="pic/banner.jpg" />
 </div>
```

```
<div id="e">
<table width="575" border="1" bordercolor="#7DD5FA" cellspacing="0"><tr><td>
<table width="575px"><tr>
      <td>用户 ID</td>
      <td>用户名</td>
      <td>密码</td>
      <td>操作时间</td>
   </tr>
   <tr class="one">
      <td>001</td>
      <td>ym</td>
      <td>***</td>
      <td>2010-08-12</td>
   </tr>
   <tr class="one">
      <td>002</td>
      <td>pkh</td>
      <td>***</td>
      <td>2010-08-23</td>
   </tr>
   <tr class="one">
      <td>003</td>
      <td>lzh</td>
      <td>***</td>
      <td>2010-07-06</td>
   </tr>
</table></td></tr></table>
```

（2）编写 CSS 样式，代码如关键技术中所示。

秘笈心法

心法领悟 185：使用 CSS 中 margin 关键字的注意事项。

（1）margin 关键字主要用于检索和设置对象四边的外沿边距，但是此关键字根据传入参数的不同，性能也不相同。如果提供全部 4 个参数值，将按上、右、下和左的顺序作用于四边；如果只提供 1 个，将用于全部的四边；如果提供 2 个，第 1 个用于上和下，第 2 个用于左和右；如果提供 3 个，第 1 个用于上，第 2 个用于左和右，第 3 个用于下。

（2）内联对象要使用该属性，必须先设定对象的 height 或 width 属性，或者设定 position 属性为 absolute。在 IE 4+中，margin 属性不可用于 td 和 tr 对象。

（3）外延边距始终透明。

实例 186	设置图片的样式 光盘位置：光盘\MR\03\186	高级 趣味指数：★★★★

实例说明

CSS 之所以受到很多 Web 设计者的追捧，是因为它几乎可以实现所有的基础美化功能，图片当然也不例外。运行本实例，应用 CSS 样式实现图片向左浮动、图片向右浮动和为图片添加边框的功能，运行结果如图 3.42 所示。

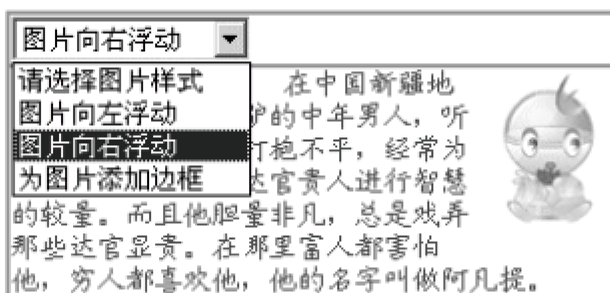


图 3.42　设置图片的样式

关键技术

本实例主要通过 JavaScript 代码动态地为图片添加 CSS 样式，核心代码如下：

```
$(document).ready(function(){
$("select").change(function(){
        var value = $(this).val();
        if(value == ""){
                alert("您没有选择图片样式");
        }else{
                if(value == 1){
                        $("img").css("float","left");
                }else{
                        if(value == 2){
                                $("img").css("float","right");
                        }else{
                                $("img").css({border:"1px red solid"});
                        }
                }
        }
});
});
```

设计过程

（1）创建脚本文件，利用<div>建立网页框架，引入 CSS 样式和 JavaScript 脚本，编写下拉列表和内容主体，代码如下：

```
<div>
<table border="1" bordercolor="#00CCFF" cellspacing="0"><tr><td>
<select>
        <option value="">请选择图片样式</option>
        <option value="1">图片向左浮动</option>
        <option value="2">图片向右浮动</option>
        <option value="3">为图片添加边框</option>
</select>
</td></tr><tr><td>
        <img src="pic/img.gif" />
            很久很久以前，  在中国新疆地区，有一个骑着毛驴的中年男人，听说他智慧过人，爱打抱不平，经常为了老百姓的利益与达官贵人进行智慧的较量。而且他胆量非凡，总是戏弄那些达官显贵。在那里富人都害怕他，穷人都喜欢他，他的名字叫做阿凡提
</td></tr></table></div>
```

（2）编写 JavaScript 脚本，代码如关键技术中所示。

秘笈心法

心法领悟 186： CSS 中如何使用 float 属性。

float 属性指定对象是否浮动和如何浮动等，参数如下：

nne：对象不浮动。

left：对象浮在左边。

right：对象浮在右边。

实例 187	滚动条样式 光盘位置：光盘\MR\03\187	高级 趣味指数：★★★★

实例说明

用户在设计页面时，为了达到整体协调的效果，很多时候需要重新设计滚动条。本实例只是更改了滚动条相关的色调，没有对其整体效果进行更改，运行效果如图 3.43 所示。



图 3.43　滚动条样式

关键技术

本实例通过 CSS 样式，对浏览器的滚动条等颜色进行更改，核心代码如下：

```
td{font-size:9pt}
BODY {
SCROLLBAR-BASE-COLOR: #FFAD00;
SCROLLBAR-FACE-COLOR: #FFCB29;
SCROLLBAR-ARROW-COLOR: #800529;
SCROLLBAR-TRACK-COLOR: #EEFFCC;
SCROLLBAR-DARKSHADOW-COLOR:#FFFFFF;
}
```

设计过程

（1）创建 index.php 文件，设置网页页面，载入 CSS 样式，其关键代码如下：

```
<link href="css/style.css" rel="stylesheet">
```

（2）编写 CSS 样式文件 style.css，代码如关键技术中所示。

秘笈心法

心法领悟 187：　CSS 中设置 color 属性的注意事项。

color 为要设置的颜色代码，可以是十六进制的，例如#FF0000，也可以是以 RGB 表示的，例如 RGB（255，0，255）。在设置滚动条样式时，不必把所有的属性都用上，用户可根据网页的需求自行设置。

实例 188	设置表单样式 光盘位置：光盘\MR\03\188	高级 趣味指数：★★★★

实例说明

表单几乎是所有网站都不可或缺的一个模块，用户可以用它来登录到网站或者填写个人信息等。所以对于一个 Web 开发者来说，学习设置一个优秀的表单样式是很重要的。运行本实例，通过 CSS 设置表单样式，如图 3.44 所示。



图 3.44　设置表单样式

关键技术

本实例通过 CSS 样式对表单的字体、标签、文本边框进行设置，实现美化的目的，代码如下：

```
body{
margin:0px;
}
div{
margin-top:150px;
}
#one{
margin-left:400px;
font-family:"楷体_GB2312";
font-size:16px;
}
input{
width:150px;
}
#e{
width:35px;
background-image:url(../pic/buttom.jpg);
border:0;
}
#ee{
width:35px;
background-image:url(../pic/buttom1.jpg);
border:0;
}
#o,#n{
border:1px #FFCCFF solid;
}
table{
width:250px;
height:150px;
background-image:url(../pic/bg.jpg);
}
```

另外还应用 JavaScript 特效，使鼠标覆盖或移出时文本框变色，代码如下：

```
$(document).ready(function(){
$("input:lt(2)").mouseover(function(){
        $(this).css("background","#BFAFC9");
        $(this).mouseout(function(){
                $(this).css("background","white");
        });
});
});
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式、JavaScript 脚本文件和 jQuery 类库文件，通过<div>定位技术，对表格信息进行定位。创建 form 表单，设置用户名和密码文本框，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<div>
<div id="one">
<table border="1" bordercolor="#00CCFF" cellspacing="0"><tr><td>
用户名：<input type="text" id="o" /><br>
密  码：<input type="password" id="n" /><br>
<input type="button" id="e" value=""/><input type="button" id="ee" value=""/>
</td></tr></table>
</div>
</div>
```

（2）编写 CSS 样式和 JavaScript 脚本，代码如关键技术中所示。

秘笈心法

心法领悟 188：CSS 中设置 border 属性的注意事项。

border 属性是复合属性，如使用该复合属性定义单个参数，则其他参数的默认值将无条件覆盖各自对应的单个属性设置，默认值为 medium none。border-color 的默认值将采用文本颜色。要使用该属性，必须先设定对象的 height 或 width 属性，或者设定 position 属性为 absolute。

实例 189　设置超链接的样式

光盘位置：光盘\MR\03\189

高级

趣味指数：★★★★

实例说明

超链接在 Web 开发中也是很常见的，其主要作用是执行客户端的脚本跳转。默认情况下，单击之前颜色为蓝色，存在下划线样式；单击之后颜色为紫色，同样存在下划线样式。但是这种样式不可能适合所有的网站色调。本实例介绍如何更改超链接的样式，运行结果如图 3.45 所示。



图 3.45　设置超链接的样式

关键技术

本实例通过 CSS 样式，定义<a>标签的 3 个属性，控制超链接的样式，其核心代码如下：

```
a{
 border:1px #FF0000 solid;
}
a:link{
 color:#333333;
 text-decoration:none;
}
a:visited{
 color:#666666;
 text-decoration:none;
}
a:hover{
 background-color:#FF0000;
 color:#FFFFFF;
}
```

设计过程

（1）创建 index.php 文件，载入 CSS 样式文件，编写 5 个超链接，代码如下：

```
<div>
 <div id="one">
 <table border="1" bordercolor="#00CCFF" cellspacing="0" background="pic/bg.jpg"><tr><td>
 <a href="#">链接 1</a> <a href="#">链接 2</a> <a href="#">链接 3</a> <a href="#">链接 4</a> <a href="#">链接
4</a> <a href="#">链接 5</a>
 </td></tr></table>
 </div>
</div>
```

（2）编写 CSS 样式，代码如关键技术中所示。

秘笈心法

心法领悟 189：定义<a>标签的 3 个属性。

（1）:link：设置 a 对象在未被访问前的样式表属性。

（2）:visited：设置 a 对象在其链接地址已被访问时的样式表属性。

（3）:hover：设置对象在鼠标悬停时的样式表属性。

实例 190 设置 body 的样式

光盘位置：光盘\MR\03\190

高级

趣味指数：★★★★

实例说明

HTML 将整个网页分成头部<head>和身体<body>两个部分，头部一般包含标题和引入的 JavaScript 和 CSS 样式等；身体部分主要是执行的代码，其中可以包含表格、表单和<div>标签等元素。大家都知道，对于<body>中的元素可以进行 CSS 的样式美化，但是如果用户想定义类似全文的一个文字的样式，则需要对每个标签里输出到页面的文字进行 CSS 样式的定义，这样未免太过麻烦，因此可以在<body>标签中应用 CSS 样式，这样就可以对整个页面中的文字进行设置，运行结果如图 3.46 所示。



图 3.46　设置 body 的样式

关键技术

本实例通过定义<body>标签的 CSS 样式，对整个页面中的文字及文字的颜色进行定义，其核心代码如下：

```
body{
 font-family:"楷体_GB2312";
 font-size:14px;
 color:#666666;
}
```

设计过程

（1）创建 index.php 文件，利用<table>标签建立网页框架，引入 CSS 样式和 JavaScript 脚本，编写表单，代码如下：

```
<link type="text/css" rel="stylesheet" href="css/in.css"/>
<script type="text/javascript" src="../js/jquery-1.3.2.js"></script>
<script type="text/javascript" src="js/in.js"></script>
<table width="595px" height="448px" align="center" border="1" bordercolor="#0099FF" cellspacing="0" background="pic/bg.jpg">
  <tr>
    <td><table align="center">
  <tr>
    <td>用户名：<input type="text" id="o" /></td>
  </tr>
  <tr>
    <td>密  码：<input type="password"  id="o"/></td>
```

```
  </tr>
</table>
</td>
  </tr>
</table>
```

（2）编写 CSS 样式和 JavaScript 脚本文件，核心代码如关键技术中所示。

秘笈心法

心法领悟 190：定义<body>标签属性的注意事项。

当用 CSS 样式定义了<body>标签后，如果想定义<body>内的元素属性，需要对每个元素进行更改。

实例 191	模拟的进度条 光盘位置：光盘\MR\03\191	高级 趣味指数：★★★★

实例说明

在一些大型网站的首页中，为了体现动画效果，通常在进入首页前添加一个模拟的进度条，通过该进度条实现打开网站首页的前奏。运行本实例，将自动加载一个模拟的进度条，运行结果如图 3.47 所示。当进度条完成 100%后，自动跳转到 http://www.mingrisoft.com 网站。



图 3.47　模拟的进度条

关键技术

本实例主要应用 for 循环语句控制进度条的进度，当进度条完成整个进度的显示时，自动调用 finish()函数，从而跳转到指定的网站。

另外，本实例中进度条的样式是通过 CSS 样式设置的，并应用 div 层输出到浏览器。

设计过程

（1）应用 CSS 样式设置 div 层及文字的样式，代码如下：

```
<style type="text/css">
div{font-size: 9pt}
#load{font-size: 9pt; cursor: default; position: absolute; display:block;width:402;height:20;top:expression((document.body.clientHeight-50)/2);
left:expression((screen.width-400)/2); display: block; z-index:100;background:#EDECE9}
.px1{border: 1px solid ;background-color: #FFFFFF;}
body {
 background-color: #000000;
}
.style1 {color: #FFFF00}
</style>
```

（2）在<body>中应用 div 层设计进度条，应用 class 为 div 层引用已定义的 CSS 样式，并添加部分说明文字，然后，应用 for 循环语句限制进度条完成整个进度的行程，代码如下：

```
<body onload=finish()>
<div id=load style="background-color:#000000"><span class="style1">请稍候...</span>
<div class=px1><img   id=line style="background:red" width=14 height=14></div>
<div class="style1" id=load_txt>loading</div>
</div>
<?php
for($i=0;$i<500;$i++){
?>
```

```
<script>load_num("Loading");</script>
<?php
}
?>
</body>
```

（3）应用 JavaScript 脚本实现进度条的动态效果，当进度条完成整个进度的显示时，自动调用 finish()函数，从而跳转到指定的网站，代码如下：

```
<script language=JavaScript>
var load_line_i=1;
var load_line_step=500;
function load_num(txt)
{load_line_i+=400/load_line_step;
load_txt.innerText=txt+" "+Math.floor(load_line_i/4.00)+"%";
document.all("line").width=load_line_i;}
function finish(){
window.location.href="http://www.mingrisoft.com";
}
</script>
```

秘笈心法

心法领悟 191：JavaScript 脚本客户端跳转的两种形式。

当页跳转：

```
window.location.href=”http://”
```

新页跳转：

```
Window.open=”http://”
```

实例 192	<div>标签设计用户注册页面 光盘位置：光盘\MR\03\192	高级 趣味指数：★★★★

实例说明

在进行网页排版时，用户可以使用<table>标签进行布局，但是在比较大的网页中，用<table>标签布局会使页面代码变得混乱，不容易解读，所以笔者建议尽量使用<div>标签进行页面布局。运行本实例，如图 3.48 所示。



图 3.48　<div>标签设计用户注册页面

关键技术

本实例的关键点是对<div>标签定位的灵活运用，及运用 CSS 样式控制页面的样式，核心代码如下：

```
body{
margin:0px;
text-align:center;
}
```

```
.one{
margin-top:100px;
width:595px;
height:448px;
background:url(../pic/bg.jpg);
}
input{
width:150px;
height:18px;
border-left:0;
border-top:0;
border-right:0;
border-bottom:1px #CCCCCC solid;
background:#6BACCC;
}
.two{
margin-top:200px;
margin-left:200px;
font-family:"楷体_GB2312";
font-size:16px;
color:#0000FF;
}
#pic{
border:0;
background:url(../pic/bt.jpg);
width:39px;
}
.three{
margin-right:170px;
}
```

设计过程

（1）创建脚本文件，利用<div>标签对网页进行排版，引入 CSS 样式和 JavaScript 脚本，编写表单，代码如下所示：

```
<div class="one">
<div class="two">
        <div>
                昵称：<input type="text" />
        </div>
        <div>
                密码：<input type="password" />
        </div>
        <div>
                QQ：<input type="text" />
        </div>
        <div>
                住址：<input type="text" />
        </div>
        <div class="three">
                <input id="pic" type="submit" value="" />
        </div>
</div>
</div>
```

（2）编写 CSS 样式，设置文本标签和标签位置定位，代码如关键技术中所示。

秘笈心法

心法领悟 192：如何定义文本框只显示为一条下滑线的 CSS 样式风格。

使文本框只显示为一条下滑线的操作很简单，其原理是将文本框的左、右、上边框的 border 属性都设置为 0，然后定义下边框的 CSS 样式。以本实例为例，代码如下：

```
border-left:0;
border-top:0;
border-right:0;
border-bottom:1px #CCCCCC solid;
```



实例 193 <div>标签设计论坛帖子浏览页面

光盘位置：光盘\MR\03\193

高级

趣味指数：★★★★

实例说明

论坛的浏览页面，相信对于所有用户来说并不陌生，其主要功能是将用户发表的内容供所有人或部分人浏览。本实例讲解的主要内容并不是如何实现浏览这一功能而是对浏览页面的<div>布局。运行本实例，如图 3.49 所示。

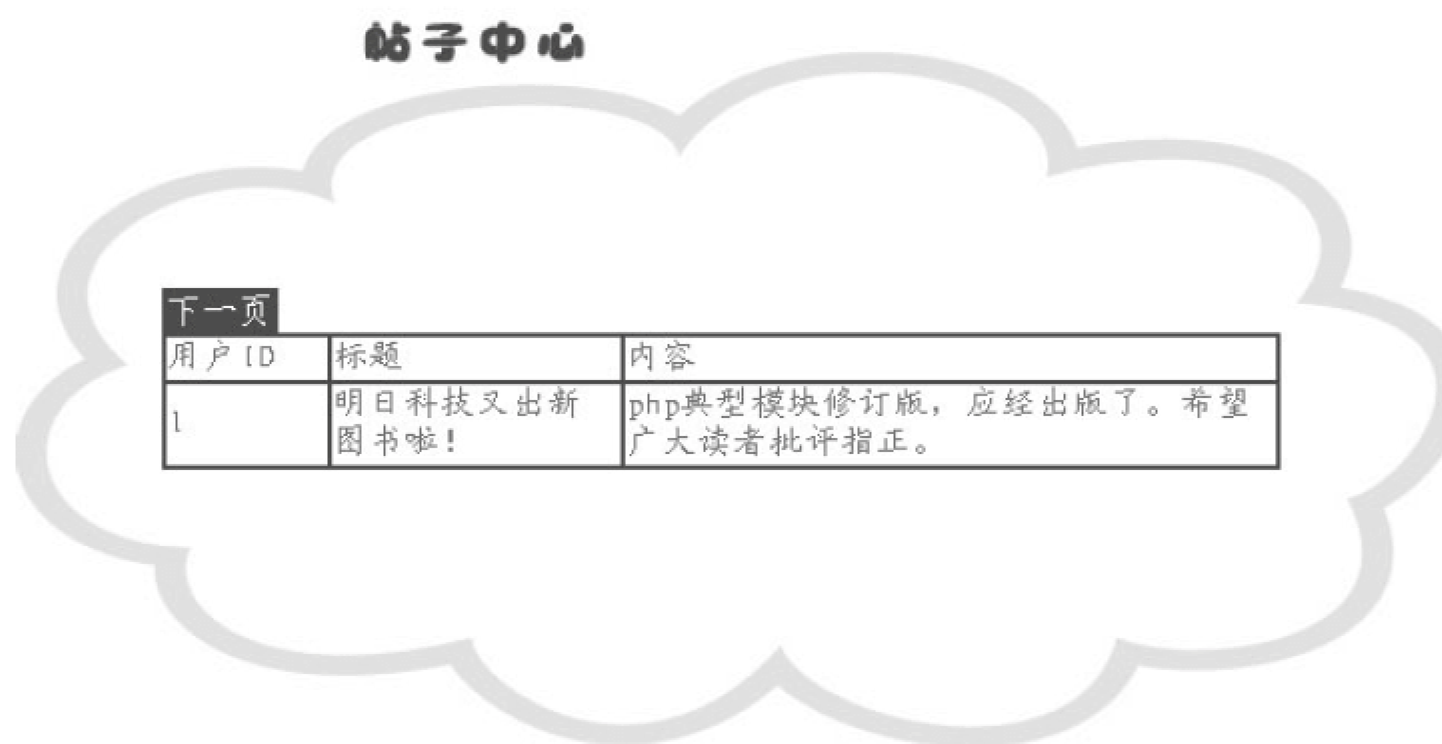


图 3.49　论坛帖子的浏览页面

关键技术

本实例的关键点是利用<div>的定位技术对页面进行合理布局，其中实现定位的 div 元素控制的代码如下：

```
.one{
margin:auto;
width:700px;
height:500px;
background:url(../pic/bg.jpg);
}
.three{
margin-top:200px;
margin-left:100px;
}
.five{
margin-left:100px;
}
```

设计过程

（1）创建脚本文件，引入 CSS 样式，连接数据库，完成数据库中数据的分页输出，代码如下：

```
<div class="one">
<div class="two">
<?php
$conn = mysql_connect("localhost","root","111") or die ("Connect MySQl False");          //连接数据库
mysql_select_db("db_database03",$conn) or die ("Connect db False");
mysql_query("SET NAMES UTF8");
$num = 1;
$rs = mysql_query("select * from tb_demo01");
$mu = mysql_num_rows($rs);
if(isset($_GET[page])){
```

```
        $page = $_GET[page];
}else{
        $page = 1;
}
if($page < $mu){                                                    //分页
?>
<div class="three">
<a href="index.php?page=<?php echo ($page+1);?>">下一页</a>
</div>
<?php
}else{
?>
<div class="three">
<a href="index.php?page=<?php echo ($page-1);?>">上一页</a>
</div>
<?php
}
$pag = $page - 1;
$sqlstr = "select * from tb_demo01 limit $pag,1";
$rss = mysql_query($sqlstr);
$rey = mysql_fetch_array($rss);
?>
<div class="five">
<table width="500px" border="1" bordercolor="#FF0000" cellspacing="0">        //数据输出
  <tr>
    <td width="70px">用户 ID</td><td width="150px">标题</td><td>内容</td>
  </tr>
  <tr>
    <td><?php echo $rey[id]?></td><td><?php echo $rey[title]?></td><td><?php echo $rey[content]?></td>
  </tr>
</table>
</div>
</div>
```

（2）编写 CSS 样式，代码如下：

```
body{
margin:0px;
}
.one{
margin:auto;
width:700px;
height:500px;
background:url(../pic/bg.jpg);
}
.three{
margin-top:200px;
margin-left:100px;
}
td{
font-family:"楷体_GB2312";
font-size:15px;
color:#0099FF
}
.five{
margin-left:100px;
}
a{
font-family:"楷体_GB2312";
border-bottom:0;
border-left:2px #FF0000 solid;
border-right:2px #FF0000 solid;
border-top:2px #FF0000 solid;
}
a:link{
text-decoration:none;
color:#000000;
```

```
}
a:visited{
text-decoration:none;
color:#000000;
}
a:hover{
background-color:#FF0000;
color:#FFFFFF;
}
```

秘笈心法

心法领悟 193：什么是相对路径与绝对路径。

相对路径与绝对路径是在日常程序开发过程中经常用到的两个概念。相对路径是指以当前文件为参照物，其他文件的位置；绝对路径，顾名思义，指文件的绝对位置，即文件在本机中的绝对位置。以本实例为例，以 background:url(../pic/bg.jpg);引入背景图片，此路径为相对路径。

实例 194 <div>标签设计论坛后台管理页面

光盘位置：光盘\MR\03\194

高级

趣味指数：★★★★

实例说明

所有的 Web 开发者几乎都是从最简单的 HTML 语言开始学起，所以较<div>标签来说，程序员更喜欢用<table>标签来完成网页框架的布局。但是从网站优化的角度来说，<div>标签确实优于<table>标签。本实例通过设计后台管理页面，着重阐述如何使用<div>布局。运行本实例，如图 3.50 所示。



图 3.50　后台管理页面

关键技术

本实例利用<div>标签进行网页布局，并利用 CSS 定义宽、高等属性。实现定位的核心代码如下：

```
body{
margin:0px;
font-family:"华文彩云";
font-size:16px;
color:#000000;
}
.a{
margin:auto;
```

```
width:1023px;
height:600px;

}
.b{
margin:0px;
width:1023px;
height:80px;
background:url(../pic/top.jpg);
}
.c{
margin:0px;
width:165px;
height:310px;
position:absolute;
background:url(../pic/left.jpg);

}
.d{
margin:0;
width:845px;
height:300px;
background:url(../pic/right.JPG);
float:right;
}
.e{
background:url(../pic/down.JPG);
width:1023px;
height:10px;
}
```

设计过程

（1）创建 index.php 文件，利用<div>标签对网页进行布局，代码如下：

```
<div class="a">
<div class="b"></div>
<div class="c">      </div>
<div class="d">
</div>
<div class="e"></div>
</div>
```

（2）编写 CSS 样式，代码如关键技术中所示。

秘笈心法

心法领悟 194：CSS 样式中，如何使文本或标签居中显示。

在<div>标签定位中，如果想使<div>标签内部的内容或标签居中显示，可以使用 CSS 中的 text-align 属性实现。

实例 195	<div>标签设计页面布局 光盘位置：光盘\MR\03\195	高级 趣味指数：★★★★

实例说明

在设计网页的过程中，能否控制好各个模块的位置是很重要的，而控制这些模块的标签中<div>是最常用的标记。利用<div>标签再加上 CSS 对其进行样式控制，可以实现很多效果。下面就来讲述一下<div>标签的使用特点和应用技巧。运行本实例，如图 3.51 所示。



图 3.51　利用<div>标签布局

关键技术

本实例用到的属性包括 margin、width、height、float、clear，具体说明如下：

（1）margin：控制模块与外边距的距离。其中，如果只定义 1 个参数，表示到四边的边距；定义 3 个参数表示到上、左/右、下的边距；定义 2 个参数表示到上/下和左/右的边距。

（2）width：定义<div>模块的宽度。

（3）hight：定义<div>模块的高度。

（4）float：指定<div>对象模块如何浮动，其中包括 3 个参数：left、right、none，分别表示向左浮动、向右浮动、不浮动。

（5）clear：指定是否允许浮动对象，其中包括 4 个参数：left、right、both、none，分别表示清除左侧浮动、清除右侧浮动、清除两侧浮动、不清除浮动。

设计过程

（1）创建脚本文件，利用<div>定位技术设计网页结构，代码如下：

```
<div class="a">
        <div class="b">
                <div class="f"></div>
                <div class="j"></div>
        </div>
        <div class="c"></div>
        <div class="d">
                <div class="g"></div>
                <div class="h"></div>
                <div class="i"></div>
        </div>
        <div class="e">
                <div class="k"></div>
        </div>
</div>
```

（2）编写 CSS 样式，将<div>模块定位到指定位置并添加背景颜色，代码如下：

```
body{
 margin:0px;
 background:#CCCCCC;
```

```
}
.a{
margin:auto;
width:700px;
height:600px;
background:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:400px;
background:#6699FF;
}
.g{
margin:5px;
width:100px;
height:400px;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:300px;
height:400px;
background:#990000;
float:left;
}
.i{
margin:5px;
width:275px;
height:400px;
background:#99FF33;
float:right;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
```

```
width:200px;
height:90px;
background:#FFFF00;
}
```

秘笈心法

心法领悟 195：clear 属性的具体应用。

假设外部<div>标签定义宽度为 400 个像素，内部定义的两个<div>标签的总宽度为 450 个像素，如果内部右侧的<div>标签不能清除左侧的浮动的话，将会产生换行效果。

实例 196	页面布局（2 列左窄右宽高度自适应） 光盘位置：光盘\MR\03\196	高级 趣味指数：★★★★

实例说明

在比较小的博客中，网页一般分为上、中、下 3 个部分，中间部分又分左、右两个部分。如果利用 HTML 的<table>标签实现，则需要多个<table>标签进行嵌套，但是如果利用<div>标签，结构就比较明朗了。下面来实现这一实例，运行结果如图 3.52 所示。

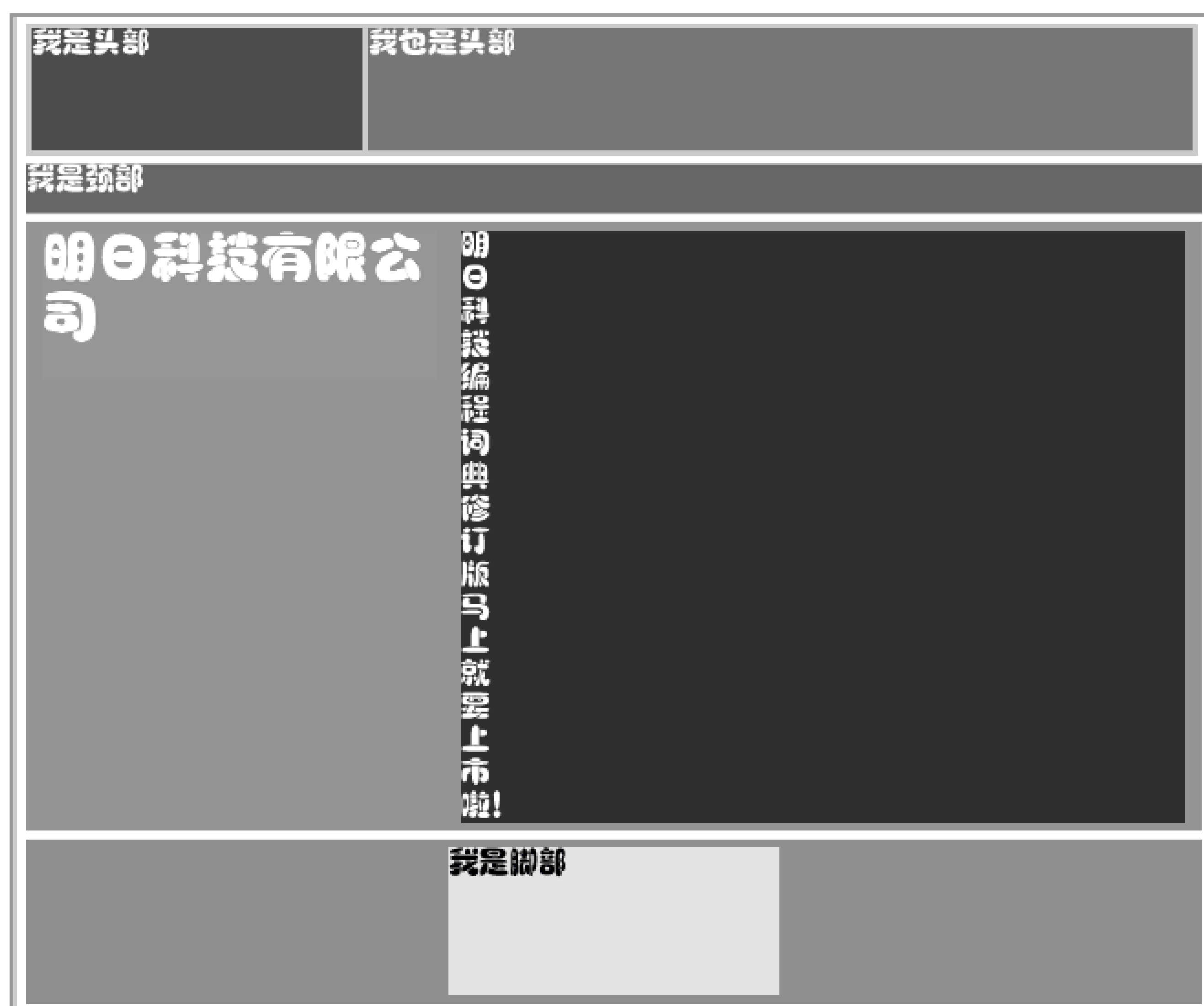


图 3.52　页面布局之 2 列左窄右宽高度自适应

关键技术

本实例的关键点是如何实现<div>模块的高度自适应。其实实现这一功能很简单，只需要定义 height 属性为 auto 即可，代码如下：

```
height:auto;
```

设计过程

（1）创建脚本文件，利用<div>标签对网页进行布局，并输出提示文字，代码如下：

```
<div class="a">
        <div class="b">
```

```
                <div class="f"></div>
                <div class="j"></div>
        </div>
        <div class="c"></div>
        <div class="d">
                <div class="g">
                        明日科技<br>PHP 开发实录<br>典型模块大全<br>
                </div>
                <div class="h">
                        明<br>日<br>科<br>技<br>编<br>程<br>词<br>典<br>修<br>订<br>版<br>马<br>上<br>就<br>要<br>上<br>市<br>啦！
                </div>
        </div>
        <div class="e">
                <div class="k"></div>
        </div>
</div>
```

（2）编写 CSS 样式，定义<div>模块的宽度和高度，并为输出的文字定义样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:600px;
background:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:400px;
background:#6699FF;
}
.g{
margin:5px;
width:200px;
height:auto;
background:#00FF00;
float:left;
font-family:"楷体_GB2312";
font-size:16px;
}
```

```
.h{
margin:5px;
width:485px;
height:auto;
background:#990000;
float:right;
clear:left;
font-family:"楷体_GB2312";
color:#FFFFFF;
font-size:18px;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
}
```

秘笈心法

心法领悟 196：什么是高度自适应。

所谓的高度自适应是指根据文本内容的多少自动调整相应的高度，这样设置的优点是，可以完全不浪费网页空间，而且使页面效果饱满丰富。高度自适应在网页设计中被广泛使用。

实例 197　页面布局（2 列固定宽度+头部+导航+尾部）

光盘位置：光盘\MR\03\197

高级

趣味指数：★★★★

实例说明

实例 196 已经讲解了页面布局中左窄右宽的情况，本实例介绍 2 列固定宽度的实现方法。运行本实例，如图 3.53 所示。

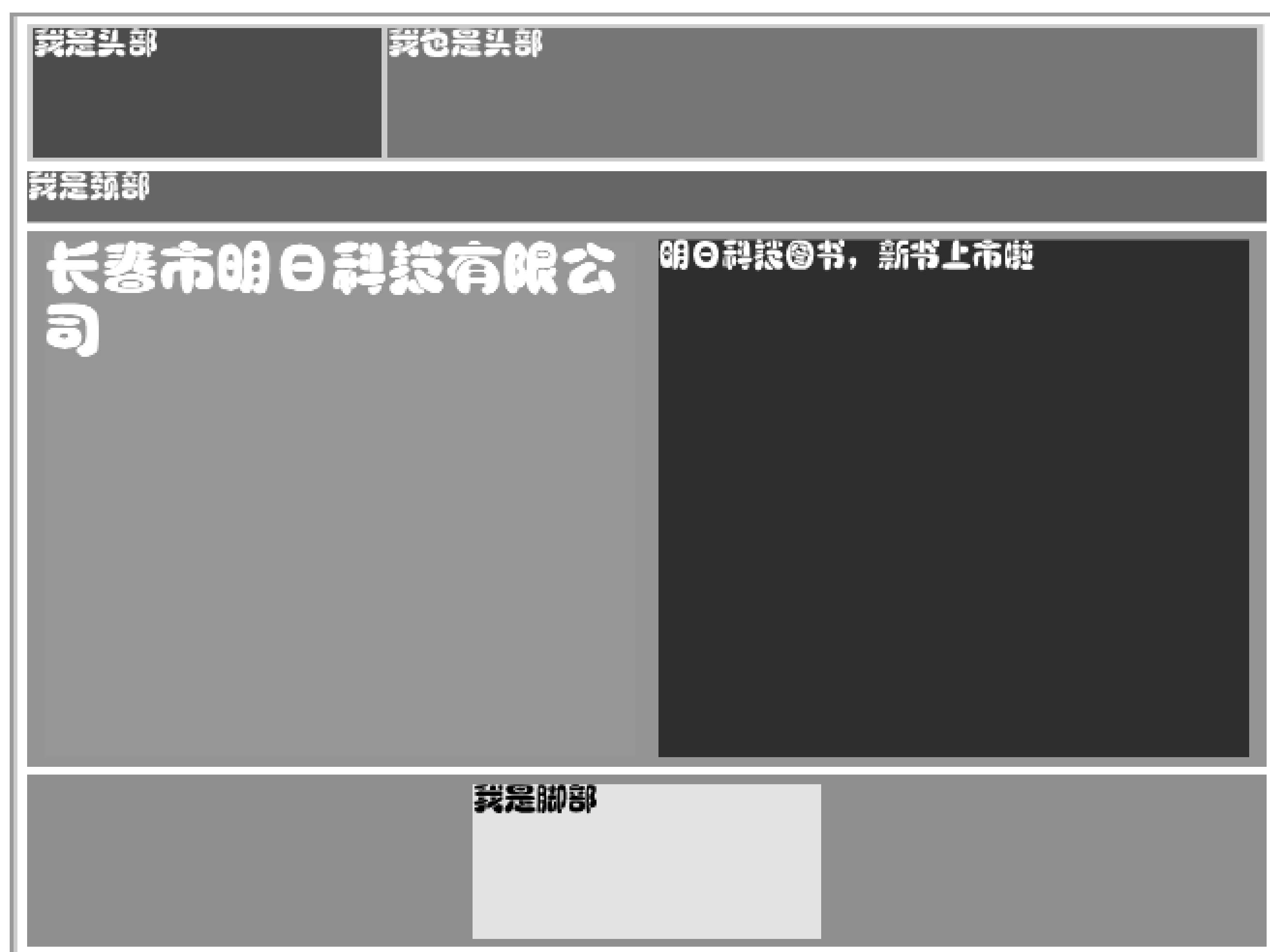


图 3.53　页面布局之 2 列固定宽度+头部+导航+尾部

关键技术

本实例设定固定宽度的方法是定义 width 属性为固定值，代码如下：

```
.g{
margin:5px;
width:340px;
height:300px;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:340px;
height:300px;
background:#990000;
float:right;
clear:left;
}
```

设计过程

（1）创建脚本文件，利用<div>标签对网页进行布局，并输出提示文字，代码如下：

```
<div class="a">
        <div class="b">
                <div class="f">我是头部</div>
                <div class="j">我也是头部</div>
        </div>
        <div class="c">我是颈部</div>
        <div class="d">
                <div class="g"><h1>长春市明日科技有限公司</h1></div>
                <div class="h">明日科技图书，新书上市啦</div>
        </div>
        <div class="e">
                <div class="k">我是脚部</div>
        </div>
</div>
```

（2）编写 CSS 样式，定义<div>相关属性，并定义页面输出字体的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
```

```
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px;
width:340px;
height:300px;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:340px;
height:300px;
background:#990000;
float:right;
clear:left;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000
}
```

秘笈心法

心法领悟 197：float 属性的使用方法。

在定义身体部分时，如果身体部分只有两部分，可以利用 float 属性，一侧向左浮动，一侧向右浮动，但是如果身体部分分为 3 个部分，那么定义为以左面起，前两个向左浮动，最后一个向右浮动为好。



实例 198 页面布局（3 列左右侧固定中间自适应居中） 高级

光盘位置：光盘\MR\03\198　趣味指数：★★★★

实例说明

实例 196 和实例 197 都是以中间为两个模块进行布局。本实例讲解中间为 3 个模块的实例，并且实现左右侧固定，中间自适应居中显示，运行效果如图 3.54 所示。

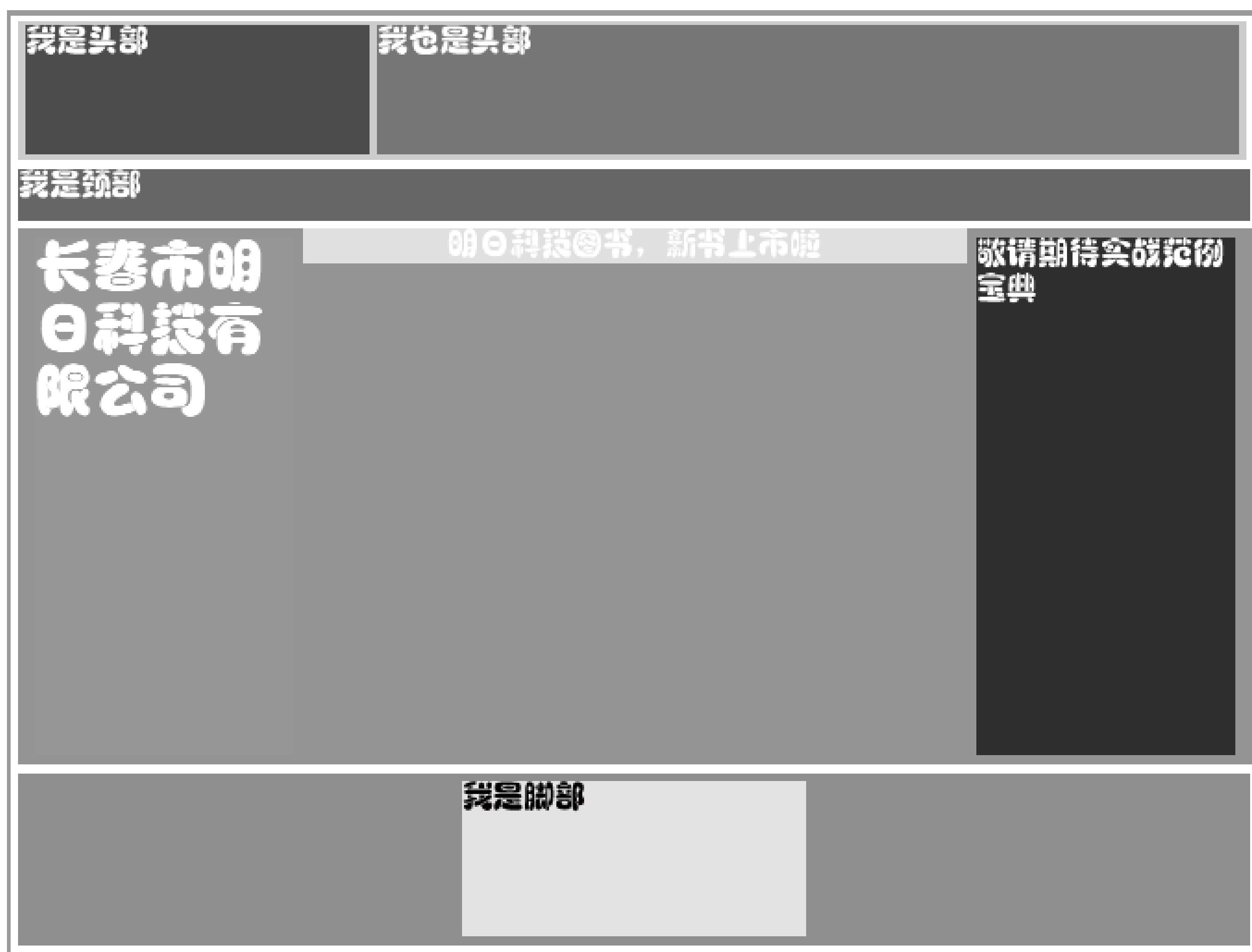


图 3.54 页面布局之 3 列左右侧固定中间自适应居中

关键技术

本实例的关键点是如何实现中间模块自适应居中。答案很简单，将模块的 margin 属性设置为上下方向与边距为 5 个像素，左右方向为 auto 自动设置即可，核心代码如下：

```
.h{
margin:5px auto;
width:385px;
height:auto;
background:#FFCCFF;
float:left;
text-align:center;
}
```

设计过程

（1）创建脚本文件，利用<div>标签进行网页定位布局，其中包括头部、颈部、身体部分和脚部，代码如下：

```
<div class="a">
<div class="b">
        <div class="f">我是头部</div>
        <div class="j">我也是头部</div>
</div>
<div class="c">我是颈部</div>
<div class="d">
        <div class="g">
                <h1>长春市明日科技有限公司</h1>
        </div>
        <div class="h">
                明日科技图书，新书上市啦
        </div>
        <div class="i">
                敬请期待实战范例宝典
        </div>
</div>
<div class="e">
        <div class="k">我是脚部</div>
</div>
</div>
```

（2）编写 CSS 样式，定义<div>标签的宽和高，并定义页面输出字体的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px;
width:150px;
height:300px;
background:#00FF00;
float:left;
}
.h{
margin:5px auto;
width:385px;
height:auto;
background:#FFCCFF;
float:left;
text-align:center;
}
.i{
margin:5px;
width:150px;
```

```
height:300px;
background:#990000;
float:right;
clear:left;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000
}
```

秘笈心法

心法领悟 198：使数据居中显示的方法。

CSS 样式中有一个 text-align 属性，相当于使用<table>标签中的 center 属性，可以使页面数据居中显示。

实例 199　页面布局（2 列右侧固定左侧自适应宽度）　高级

光盘位置：光盘\MR\03\199　趣味指数：★★★★

实例说明

灵活地控制页面中<div>标签的宽度和高度是使用<div>标签的核心。本实例实现身体部分 2 列右侧固定左侧自适应宽度，未知高度+头部+导航+尾部的页面布局，如图 3.55 所示。



图 3.55　页面布局之 2 列右侧固定左侧自适应宽度

关键技术

本实例的关键点是实现身体部分左侧自适应宽度且未知高度。实现思想是设置 width、height 属性，代码

如下：

```
.g{
margin:5px;
width:auto;
height:auto;
background:#00FF00;
float:left;
}
```

设计过程

（1）创建脚本文件，利用<div>标签建立网页的主体架构，输出文本信息，代码如下：

```
<div class="a">
<div class="b">
        <div class="f">我是头部</div>
        <div class="j">我也是头部</div>
</div>
<div class="c">我是颈部</div>
<div class="d">
        <div class="g">
                <h1>长春市明日科技有限公司</h1>
        </div>
        <div class="h">
                明日科技图书，新书上市啦
        </div>
</div>
<div class="e">
        <div class="k">我是脚部</div>
</div>
</div>
```

（2）编写 CSS 样式，定义<div>标签的具体属性，并更改输出文本的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
```

```
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px;
width:auto;
height:auto;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:320px;
height:300px;
background:#FFCCFF;
float:right;
text-align:center;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000
}
```

秘笈心法

心法领悟 199：关键字 auto 的使用。

在<div>标签中，如果不知道该标签的具体宽度和长度或者与四边的距离，可以将这些属性设置为 auto。这样一来，可以让<div>标签的长度和宽度自动调节。



实例 200　页面布局（2 列左固定右自适宽度+底平齐）　高级

光盘位置：光盘\MR\03\200　趣味指数：★★★★

实例说明

利用<div>设计页面时，有很多很小的技巧，灵活地掌握这些技巧可以使用户对页面设计豁然开朗。本实例主要实现身体部分的 2 列左侧固定右侧自适应宽度，未知高度且底部平齐+头部+导航+尾部的页面布局，运行效果如图 3.56 所示。

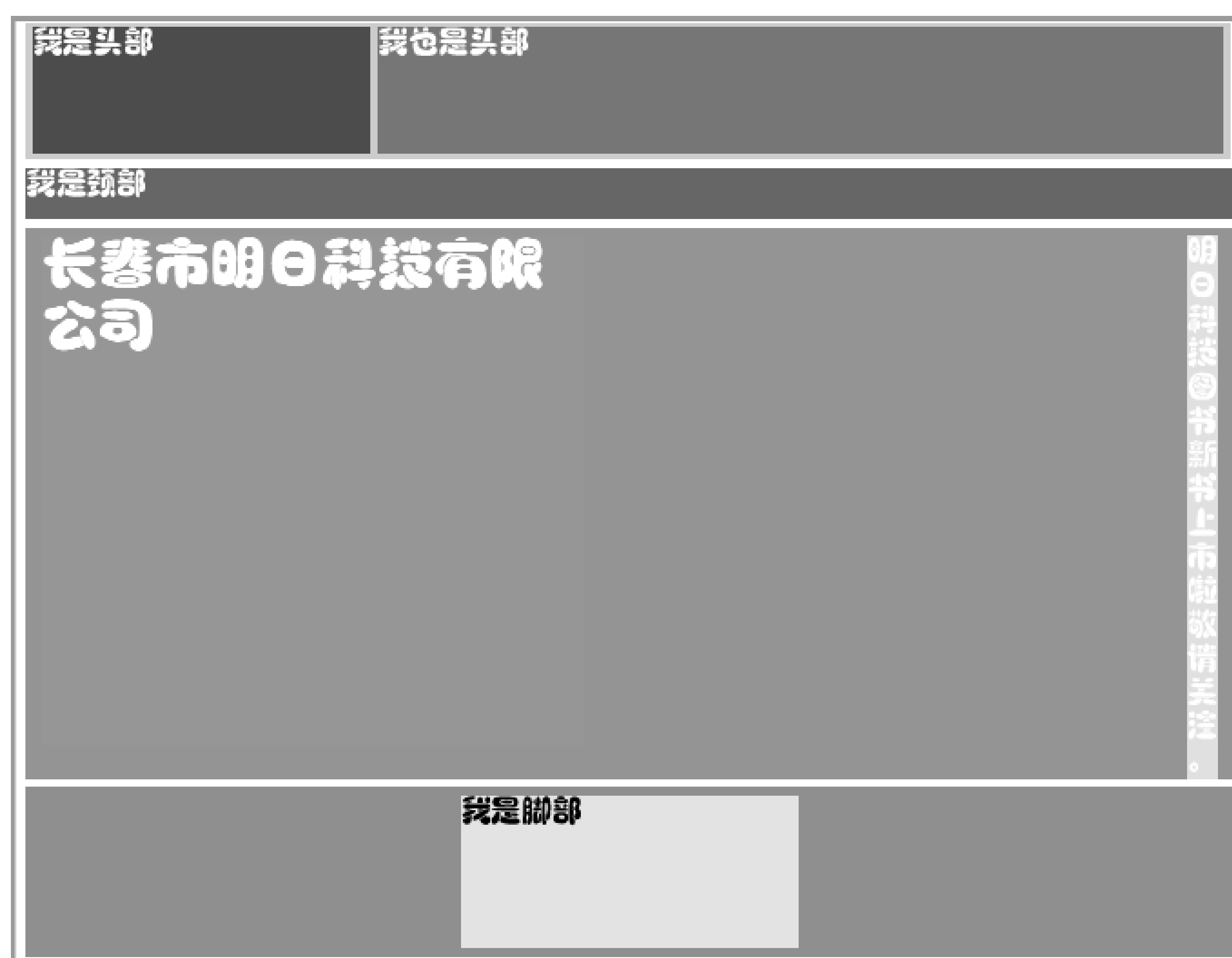


图 3.56　页面布局之 2 列左固定右自适宽度+底平齐

关键技术

本实例的关键点是实现身体部分右侧自适应宽度且未知高度，底部平齐。实现思想是设置 width、height、margin 属性，代码如下：

```
.h{
margin:5px 5px 0px 5px;
width:auto;
height:auto;
background:#FFCCFF;
float:right;
text-align:center;
}
```

设计过程

（1）创建脚本文件，利用<div>标签建立网页的主体架构，输出文本信息，代码如下：

```
<div class="a">
<div class="b">
        <div class="f">我是头部</div>
        <div class="j">我也是头部</div>
</div>
<div class="c">我是颈部</div>
<div class="d">
        <div class="g">
                <h1>长春市明日科技有限公司</h1>
        </div>
        <div class="h">
                明<br>日<br>科<br>技<br>图<br>书<br>新<br>书<br>上<br>市<br>啦<br>敬<br>请<br>关<br>注<br>
        </div>
</div>
<div class="e">
        <div class="k">我是脚部</div>
</div>
</div>
```

（2）编写 CSS 样式，定义<div>标签的具体属性，并更改输出文本的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
```

```
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px;
width:320px;
height:300px;
background:#00FF00;
float:left;
}
.h{
margin:5px 5px 0px 5px;
width:auto;
height:auto;
background:#FFCCFF;
float:right;
text-align:center;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000
}
```

秘笈心法

心法领悟 200：属性 margin 与底部平齐。

margin 属性，在实例 185 中已经讲解过，所谓的底部平齐就是指下边距为 0 像素，所以只要设置当前模块的下边距属性值为 0 像素即可。

实例 201	页面布局（2 列右固定左自适宽度+底平齐） 光盘位置：光盘\MR\03\201	高级 趣味指数：★★★★

实例说明

本实例是实例 200 的对照，意在使读者对<div>定位技术有深入的了解，运行结果如图 3.57 所示。

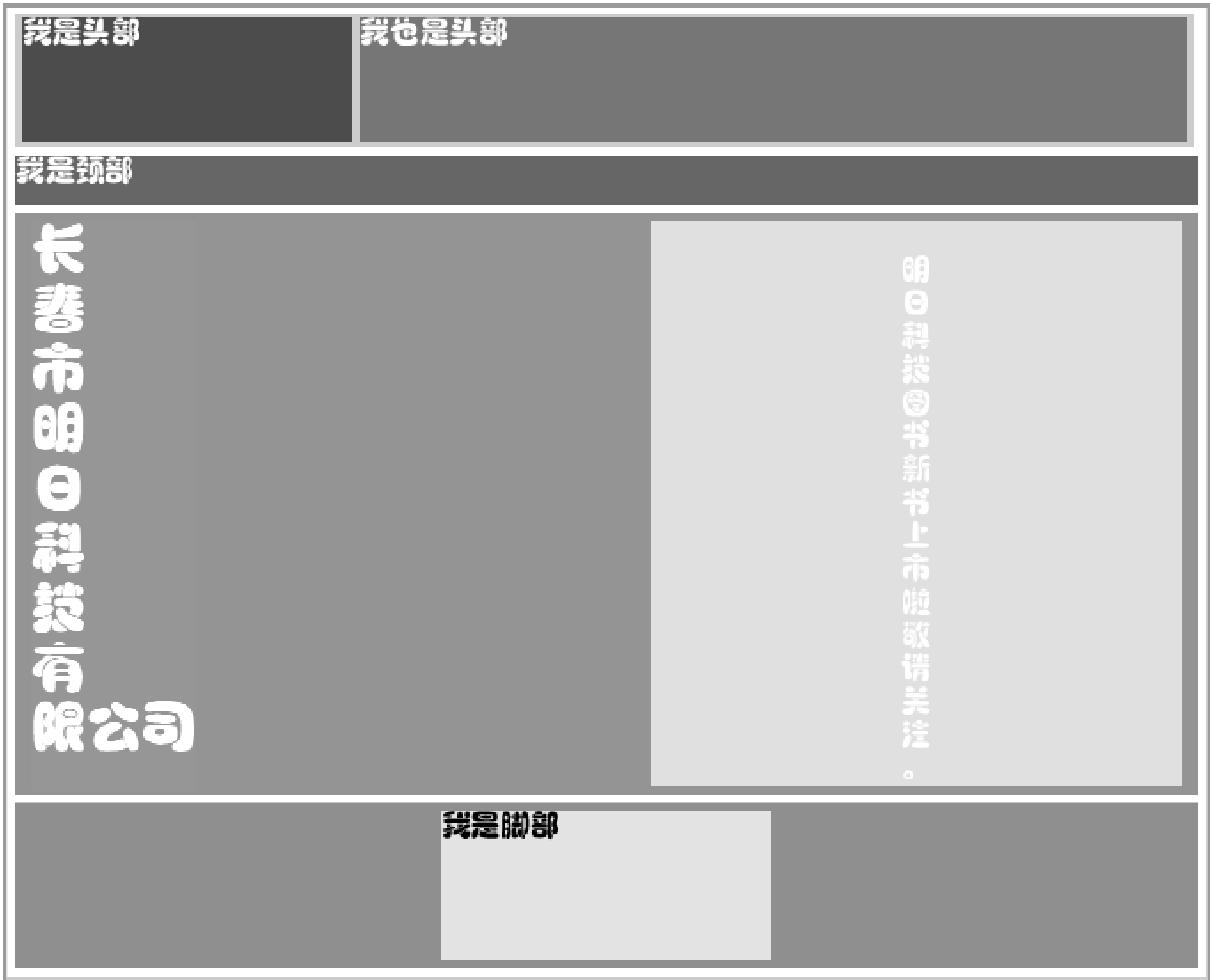


图 3.57　页面布局之 2 列右固定左自适宽度+底平齐

关键技术

本实例的方法与实例 200 相同，这里不再赘述。

设计过程

（1）创建脚本文件，利用<div>标签建立网页的主体架构，输出文本信息，代码如下：

```
<div class="a">
<div class="b">
        <div class="f">我是头部</div>
        <div class="j">我也是头部</div>
</div>
<div class="c">我是颈部</div>
<div class="d">
        <div class="g">
                <h1>长春市明日科技有限公司</h1>
        </div>
        <div class="h">
```

```
明<br>日<br>科<br>技<br>图<br>书<br>新<br>书<br>上<br>市<br>啦<br>敬<br>请<br>关<br>注<br>
        </div>
</div>
<div class="e">
        <div class="k">我是脚部</div>
</div>
</div>
```

（2）编写 CSS 样式，定义<div>标签的具体属性，并更改输出文本的样式，代码如下：

```
body{
margin:0px;
background:#CCCCCC;
}
.a{
margin:auto;
width:700px;
height:500px;
background:#FFFFFF;
font-family:"经典趣体简";
font-size:18px;
color:#FFFFFF;
}
.b{
margin:5px;
width:700px;
height:80px;
background:#CCCCCC;
}
.f{
margin:2px;
width:200px;
height:75px;
float:left;
background:#FF0000;
}
.j{
margin:2px;
width:500px;
height:75px;
float:right;
clear:left;
background:#0099FF;
}
.c{
margin:5px;
width:714px;
height:30px;
background:#666666;
}
.d{
margin:5px;
width:714px;
height:300px;
background:#6699FF;
}
.g{
margin:5px 5px 0px 5px;
width:auto;
height:auto;
background:#00FF00;
float:left;
}
.h{
margin:5px;
width:320px;
height:300px;
background:#FFCCFF;
```

```
float:right;
text-align:center;
}
.e{
margin:5px;
width:714px;
height:100px;
background:#CC66CC;
}
.k{
margin:5px auto 5px;
width:200px;
height:90px;
background:#FFFF00;
color:#000000
}
```

秘笈心法

心法领悟 201：padding 补丁边距。

本实例中，因为设置底部平齐，所以右侧固定宽度的的模块就可能与底边距不是 5 个像素的距离，为了整体的美感效果，可以为右侧模块设置 padding 补丁边距，设置好合适距离后，显示效果如图 3.57 所示。

3.6 弹 出 窗 口

弹出窗口经常在网站中使用，如弹出式广告或网站公告，及打开居中显示详细信息的新窗口等。所以掌握设计弹出窗口的技术，也许会给设计的网站带来额外的效益。

实例 202	弹出指定尺寸的无边框窗口 光盘位置：光盘\MR\03\202	高级 趣味指数：★★★★

实例说明

JavaScript 这门脚本语言的功能很强大，尤其是在网页特效的处理上，更是有其独到之处。本实例将讲述的是通过 JavaScript 代码控制，弹出指定尺寸的无边框窗口，如图 3.58 所示。

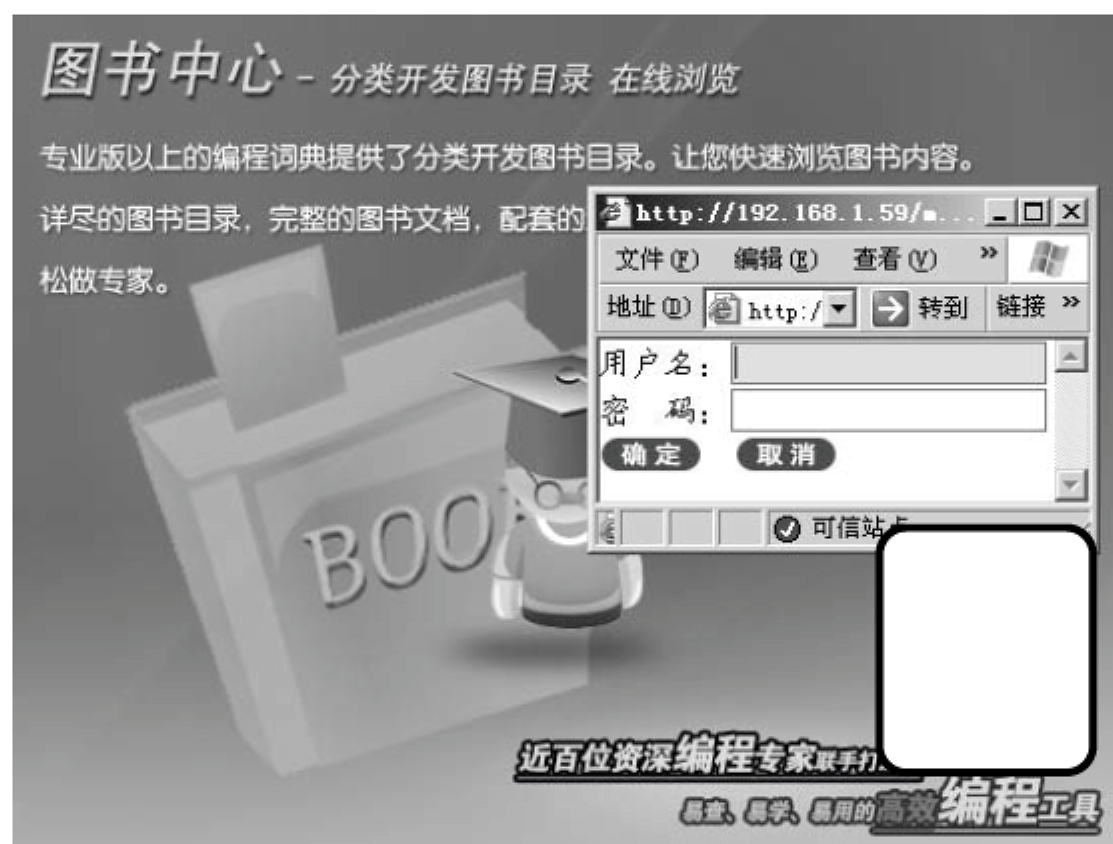


图 3.58　单击跳转图

关键技术

本实例主要是利用 JavaScript 代码，控制弹出窗口的尺寸，核心代码如下：

```
self.resizeTo(250,180);
width=screen.width;
height=screen.height;
self.moveTo((width-240)/2,(height-139)/2);
```

设计过程

（1）创建脚本文件，利用<div>标签定位技术建立网页架构，引入 CSS 文件，当单击图标按钮时，发生跳转，代码如下：

```
<div class="a"><img src="pic/bg.jpg" border="0" usemap="#Map" />
        <map name="Map" id="Map"><area shape="rect" coords="469,298,541,382" href="in.php" /></map>
</div>
```

（2）创建脚本文件，命名为 in.php，此文件执行跳转后的代码。另外，在此文件中建立<form>表单，代码如下：

```
<div class="c">
    用户名：<input type="text" name="text"><br>
    密  码：<input type="password"><br>
    <input type="button" class="one" value="">  <input type="button" class="two"value="">
</div>
```

（3）编写 JavaScript 脚本，定义网页窗口的尺寸并定义当鼠标覆盖和移开文本框时产生的变色效果，代码如下：

```
$(document).ready(function(){
self.resizeTo(250,180);                          //定义窗口的宽和高
width=screen.width;
height=screen.height;
self.moveTo((width-240)/2,(height-139)/2);
$("input:eq(0)").mouseover(function(){           //当鼠标覆盖时
    $("input:eq(0)").css("background","#FFCCFF");
    $("input:eq(0)").mouseout(function(){        //当鼠标移开时
        $("input:eq(0)").css("background","#FFFFFF");
    });
});
$("input:eq(1)").mouseover(function(){
    $("input:eq(1)").css("background","#FFCCFF");
    $("input:eq(1)").mouseout(function(){
        $("input:eq(1)").css("background","#FFFFFF");
    });
});
});
```

秘笈心法

心法领悟 202：HTML 的<map>标签。

如果用户使用的编程工具是 Dreamweaver，那么对于这个标签应该并不陌生，此标签的主要作用是当鼠标单击指定区域时，网页发生跳转。

实例 203	全屏显示无边框有滚动条窗口 光盘位置：光盘\MR\03\203	高级 趣味指数：★★★★

实例说明

通常用户在使用 IE 浏览器浏览网页时，浏览器窗口都是包括标题栏、菜单栏和状态栏等固定内容。虽然菜单栏和状态栏可以通过设置而隐藏，但是标题栏却不能，即使将网页全屏显示也不能将其去除。有时为了网站

的整体效果考虑，需要将网页设置为全屏显示。例如，在明日网络宽带影院网站中，为了给用户以更真实的感觉，就可以将整个网站的主页面设置为全屏显示模式，如图 3.59 所示，从图中可以看出该页面中还设计了“退出”超链接，可以方便浏览者随时离开影院，这也是设计全屏显示时需要注意的问题，否则浏览者会因找不到出口而烦恼。



图 3.59 全屏显示

关键技术

本实例主要通过 window 对象的 open()方法，并借助一个中转页实现，当中转页运行时调用 open()方法打开一个全屏显示的窗口（使用 window 对象的 open()方法中的参数 fullscreen=1 实现），并关闭打开中转页的窗口。

设计过程

新建一个空的 PHP 页面（页面中可以没有任何页面布局元素或内容），命名为 index.php。该页面只是起到一个页面跳转的作用，主要应用 Javascript 脚本自定义一个 MM_c()函数，并应用 window 对象的 open()方法打开一个新的 formal.htm 页，设置参数 fullscreen=1，从而实现打开窗口的全屏显示，代码如下：

```
<object type="application/x-oleobject" id=closes classid="clsid:adb880a6-d8ff-11cf-9377-00aa003b7a11">
<param name="command" value="close">
</object>
<script language="javascript">
MM_c();
function MM_c(){
        closes.Click();
        window.open("formal.htm","","fullscreen=1");
}
</script>
```

秘笈心法

心法领悟 203：全屏显示的主要应用。

用户在浏览类似于土豆、优酷等网站的视频时，经常会看到播放器的下方存在一个全屏显示的按钮，其实现方法与本实例大同小异，只不过使用的控制方法不同而已。



实例 204　网页拾色器

光盘位置：光盘\MR\03\204

高级

趣味指数：★★★★

实例说明

在网站开发过程中，用户经常要求在客户端应用自定义颜色来改变某些信息或控件等的颜色值，如留言本中的文字颜色、控件背景色等，通过在网页中加入一个网页拾色器即可实现。运行本实例，将在页面中显示如图 3.60 所示的网页拾色器，当用户单击颜色块时，就会弹出一个提示框，提示其所选的颜色值是多少，通过选择的颜色值来改变网页的背景色。

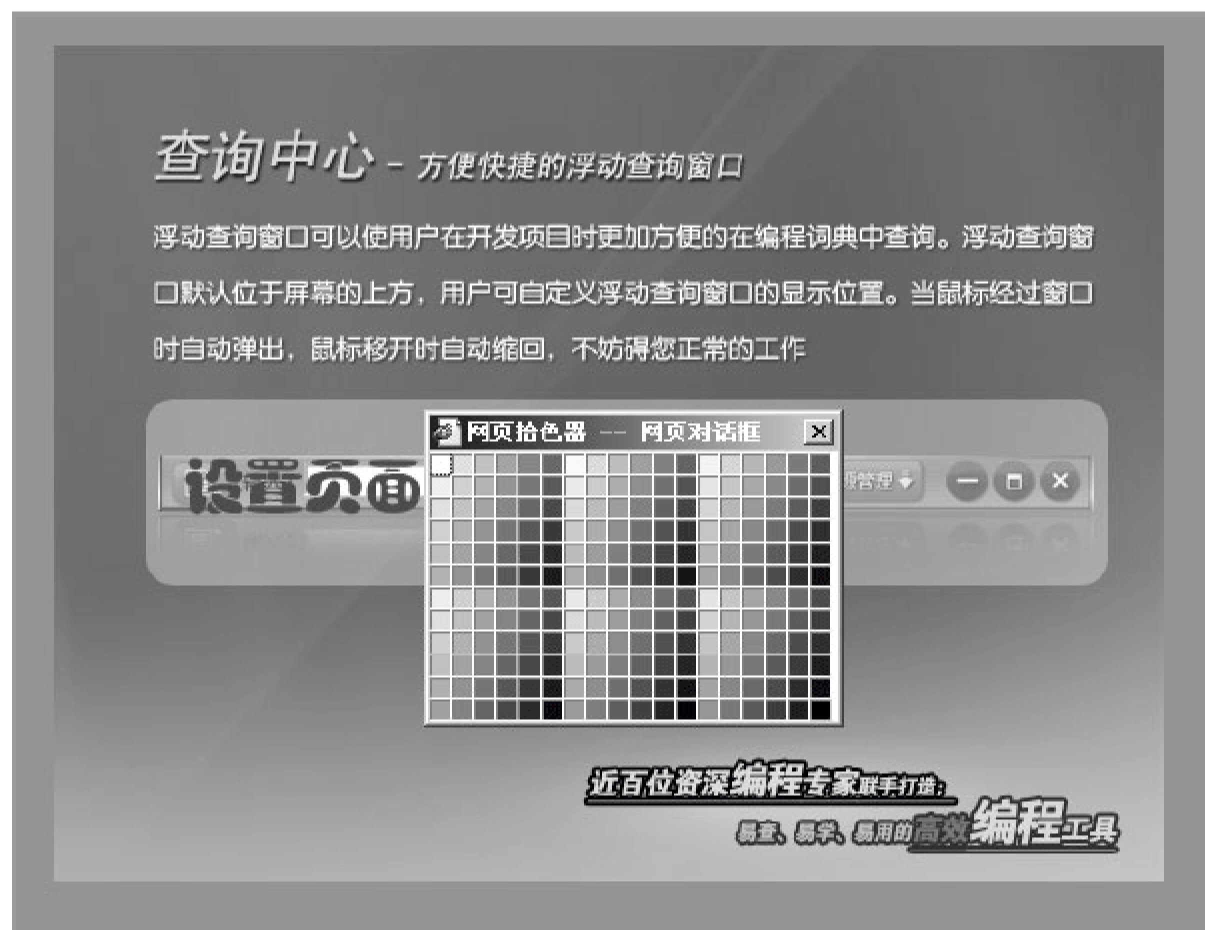


图 3.60　网页拾色器

关键技术

本实例的关键点是在实现网页拾色器时，应用 JavaScript 的数组。创建数组有以下 3 种方法：

（1）无参数调用，语法格式如下：

```
var h = new Array();
```

（2）指定数组前 n 个元素的值，语法格式如下：

```
var h = new Array(arglist);
```

其中参数 arglist 是一个用逗号隔开的值表，这些值用于给 variant 所包含的数组的各元素赋值。如果不提供参数，则创建一个长度为 0 的数组。

（3）指定具有的元素个数，语法格式如下：

```
var h = new Array(n);
```

其中参数 n 是指定数组的长度。由于在 JavaScript 中，数组的第 1 个元素的下标值为 0，所以 n 的值为数组的最大下标值加 1。

设计过程

（1）创建需要调用网页拾色器的页面，在该页面中添加一个文本框，将文本框设置为只读，同时将文本框的背景颜色设置为黑色，并在该文本框的 onClick 事件中调用自定义的 JavaScript 函数 colorpick()，代码如下：

```
<input name="color" type="text" id="color" size="3" readonly="yes" style="background-color:#CCFF00" onClick="colorpick(this);">
```

（2）编写自定义的 JavaScript 函数 colorpick()，用于打开网页对话框调用网页拾色器，并用网页拾色器的

返回值设置文本框的背景颜色和网页的背景颜色，代码如下：

```
<script language="javascript">
function colorpick(field){
    var rtn = window.showModalDialog("color.php","","dialogWidth:225px;dialogHeight:170px;status:no;help:no;scrolling=no;scrollbars=no");
    if(rtn!=null)
 document.body.style.background=rtn;
 field.style.background=rtn;
        return;
    }
</script>
```

（3）制作网页拾色器页面 color.php，关键代码如下：

```
<script language="JavaScript">
var h = new Array(6)
h[0] = "FF";
h[1] = "CC";
h[2] = "99";
h[3] = "66";
h[4] = "33";
h[5] = "00";
function action(RGB) {
  parent.window.returnValue="#"+RGB;
    window.close();
}
function Mcell(R, G, B) {
  document.write('<td bgcolor="#' + R + G + B + '">');
  document.write('<a href="#" onClick="action(\'' + (R + G + B) + '\')">');
  document.write('<img border=0 height=12 width=12 \')" alt=\'#'+R+G+B+'\'>');
  document.write('</a>');
  document.write('</td>');
}
function Mtr(R, B) {
  document.write('<tr>');
  for (var i = 0; i < 6; ++i) {
    Mcell(R, h[i], B);
  }
  document.write('</tr>')
}
function Mtable(B) {
  document.write('<table cellpadding=0 cellspacing=0 border=0>');
  for (var i = 0; i < 6; ++i) {
    Mtr(h[i], B);
  }
document.write('</table>');
}
function Mcube() {
  document.write('<table cellpadding=0 cellspacing=0 border=0><tr>');   //
  for (var i = 0; i < 6; ++i) {
    if(i%3==0){
    document.write('<tr>');
    }
   document.write('<td bgcolor="#FFFFFF">');
   Mtable(h[i])
   document.write('</td>');
  }
    if(i%3==0){
    document.write('</tr>');
    }
  document.write('</tr></table>');
}
Mcube()
</script>
```

秘笈心法

心法领悟 204：浅谈 variant 变量。

本实例中没有作为数组声明的 variant 也可以表示数组。除了长度固定的字符串以及用户定义类型之外，

variant 变量可以表示任何类型的数组。尽管一个包含数组的 variant 和一个元素为 variant 类型的数组在概念上有所不同，但对数组元素的访问方式是相同的。

实例 205　日期选择器　高级
光盘位置：光盘\MR\03\205　趣味指数：★★★★

实例说明

目前，日期型数据格式有多种，采用录入方式相对来说比较繁琐，而且采用这种方式也不利于日期格式的统一，所以可以在信息录入页面中加入一个简单的日期选择器来解决上述问题。运行本实例，在新奥家电连锁后台管理系统的销售查询页面中，单击“售货日期”文本框后的日期选择图标，会弹出“日期选择器”对话框，选择售货日期的起始日期后，单击“确定”按扭，即可成功地将选择的日期添加到对应的日期文本框中，运行结果如图 3.61 所示。



图 3.61　日期选择器

关键技术

由于日期为连续的数字，开发网页时将这些信息逐个写在<option>与</option>标记之间固然可行，但这会增加程序开发时间，开发本实例时，笔者将在下拉列表框中显示的连续数据通过 PHP 的循环结构显示出来，这样更灵活方便，实现该过程的代码如下：

```
<form name="form2">
  <tr bgcolor="#826650">
    <td width="13"><div align="center"><a href="#" onClick="addday()"></a></div></td>
    <td width="80">
<select name="n1">
<?php
 for($i=2005;$i<=2050;$i++){
?>
        <option value=<?php echo $i;?>><?php echo $i;?></option>
<?php
   }
?>
    </select>
    年</td>
    <td width="64"><select name="y1">
<?php
 for($i=1;$i<=12;$i++){
?>
        <option value=<?php echo $i;?>><?php echo $i;?></option>
  <?php
   }
```

```
?>
      </select>
      月</td>
      <td width="77"><select name="r1">
<?php
 for($i=1;$i<=31;$i++){
?>
         <option value=<?php echo $i;?>><?php echo $i;?></option>
 <?php
  }
?>
      </select>
      日</td>
      <td width="16"><div align="center"><a href="#" onClick="subday()"></a></div></td>
   </tr>
   </form>
```

设计过程

（1）在父窗口中创建函数 open_day_from()和函数 open_day_to()，实现子窗口的弹出和父窗口与子窗口之间的信息传递，代码如下：

```
<table width="250" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
 <form name="form2">
   <tr bgcolor="#826650">
      <td width="13"><div align="center"><a href="#" onClick="addday()"></a></div></td>
      <td width="80">
<select name="n1">
<?php
 for($i=2005;$i<=2050;$i++) {
?>
         <option value=<?php echo $i;?>><?php echo $i;?></option>
<?php
  }
?>
      </select>
      年</td>
      <td width="64"><select name="y1">
<?php
 for($i=1;$i<=12;$i++) {
?>
         <option value=<?php echo $i;?>><?php echo $i;?></option>
  <?php
  }
?>
      </select>
      月</td>
      <td width="77"><select name="r1">
<?php
 for($i=1;$i<=31;$i++) {
?>
         <option value=<?php echo $i;?>><?php echo $i;?></option>
 <?php
  }
?>
      </select>
      日</td>
      <td width="16"><div align="center"><a href="#" onClick="subday()"></a></div></td>
   </tr>
   </form>
</table>
<table width="250" height="20" border="0" align="center" cellpadding="0" cellspacing="0">
   <tr>
      <td bgcolor="#826650"><div align="center"><input type="button" onClick="close_day()" class="buttoncss" value="确定"></div></td>
   </tr>
</table>
```

（2）单击“确定”按钮，调用 close_day()函数实现日期的传值，代码如下：

```
<script language="javascript">
function close_day(){
window.returnValue=document.form2.n1.value+"-"+document.form2.y1.value+"-"+document.form2.r1.value;
   window.close();
}
</script>
```

秘笈心法

心法领悟 205：JavaScript 弹出窗口。

实现窗口之间的信息传递主要通过 window 的 showModalDialog()方法实现，其语法格式如下：

```
Window.showModalDialog('弹出窗口地址', '弹出窗口名称', '弹出窗口所具有的属性')
```

窗口所具有的属性包括窗口的高度、宽度，以及窗口是否有地址栏、状态栏、滚动条等一些特性。

实例 206　弹出全屏显示的网页模式对话框

光盘位置：光盘\MR\03\206

高级

趣味指数：★★★★

实例说明

为了方便用户浏览网页信息，可以将网页信息以全屏显示的网页模式展示给用户，这样一来可以方便用户浏览到网页的全部信息；二来可以保证用户查看后能及时将其关闭。运行本实例，在明日科技编程词典网站中单击“购买须知”超链接后，将弹出如图 3.62 所示的“购买须知”对话框，该页面就是通过全屏显示的网页模式对话框，用户只有关闭该页面后，才可以查看网站的其他内容。

★ 编程词典系列版本及价格 ★

● Visual Basic编程词典

软件版本	功能	享受服务	价格（元）
标准版	详细功能	一年两次升级	128
开发版	详细功能	每月升级，每月赠送最新软件，编程锦囊。全年提供30个项目源程序的有限服务支持。	568
专业版	详细功能	每月升级，每月赠送最新软件，编程锦囊。全年提供60个项目源程序的有限服务支持。	1998
企业版	详细功能	每月升级，每月赠送最新软件，编程锦囊。提供一年全程技术支持服务。	6998

● Delphi编程词典

软件版本	功能	享受服务	价格（元）
标准版	详细功能	一年两次升级	128
开发版	详细功能	每月升级，每月赠送最新软件，编程锦囊。全年提供30个项目源程序的有限服务支持。	568
专业版	详细功能	每月升级，每月赠送最新软件，编程锦囊。全年提供60个项目源程序的有限服务支持。	1998
企业版	详细功能	每月升级，每月赠送最新软件，编程锦囊。提供一年全程技术支持服务。	6998

图 3.62　弹出全屏显示的网页模式对话框

关键技术

本实例主要应用 screen 对象的 width、height 属性和 window 对象的 showModalDialog()方法实现，showModalDialog()方法用于弹出全屏显示的网页模式对话框，其语法格式如下：

```
variant = object.showModalDialog(sURL [, vArguments [, sFeatures]])
```

参数说明：

sURL：指定 URL 文件地址。

vArguments：用于向网页对话框传递参数。传递参数的类型不受限制，对于字符串类型，最大为 4096 个字

符，但也可以传递对象，例如 index.html。

设计过程

（1）创建提供“购买须知”超链接的页面，在该页面中添加空的超链接，在该超链接的 onClick 事件中调用自定义 JavaScript 函数 opendialog()，代码如下：

```
<a href="#" onClick="opendialog()">购买须知</a>
```

（2）编写自定义 JavaScript 函数 opendialog()，用于打开全屏显示的网页对话框，代码如下：

```
<script language="javascript">
 function opendialog(){
 var width=screen.width;
 var height=screen.height;
 window.showModalDialog("notice.php","","dialogWidth="+width+"px;dialogHeight="+height+"px;status=no;help=no;scrollbars=no")
 }
</script>
```

秘笈心法

心法领悟 206：showModalDialog()与 showModelessDialog()的区别是什么？

二者的区别在于 showModalDialog()打开的网页对话框为模式窗口，置在父窗口上，必须关闭才能访问父窗口；而 showModelessDialog()打开的对话框是无模式窗口，打开后不必关闭也可以访问父窗口或其他窗口。

实例 207 关闭 IE 主窗口时不弹出询问对话框

高级

光盘位置：光盘\MR\03\207

趣味指数：★★★★

实例说明

通常情况下，当使用 JavaScript 的 window.close()语句关闭 IE 主窗口时，会弹出一个“您查看的网页正在试图关闭窗口。是否关闭此窗口？”的询问对话框。在制作网络应用程序时，这种情况是用户不想看到的，那么如何才能屏蔽该对话框呢？本实例将介绍解决该问题的方法。运行本实例，如图 3.63 所示，单击“关闭窗口”超链接后，该 IE 窗口将被直接关闭。

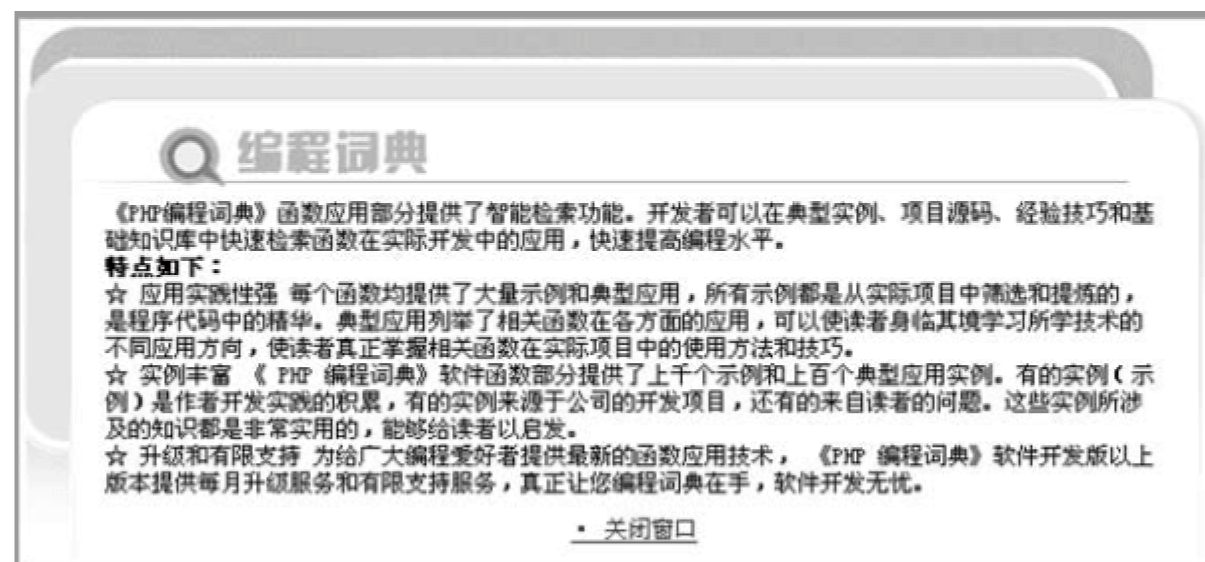


图 3.63　关闭窗口不弹出对话框

关键技术

本实例主要应用 window 对象的 opener 属性将要关闭的 IE 窗口的打开窗口设置为 null，然后再通过 window 对象的 close()方法将该窗口关闭，这样就不会弹出询问对话框。

设计过程

创建 index.php 文件，添加一个用于关闭窗口的超链接，在该超链接的 onClick 事件中添加控制窗口关闭的

代码，如下所示：

```
<a href="#" onClick="window.opener=null;window.close();">• 关闭窗口</a>
```

秘笈心法

心法领悟 207：详解 opener。

opener 是一个可读可写的属性，对一个 window 对象返回一个引用，用于指定打开本窗口的窗口对象，语法格式如下：

```
window.opener
window.opener.方法
window.opener.属性
```

opener 属性返回的是一个窗口对象，其与打开该窗口的父窗口相联系，当访问子窗口的 opener 属性时，返回的是父窗口。通过该属性，可以使用父窗口对象中的方法和属性。

实例 208	弹出提示对话框并重定向网页 光盘位置：光盘\MR\03\208	高级 趣味指数：★★★★

实例说明

在网站开发过程中，需要考虑网站的安全性，例如，设置允许哪些用户访问是一项非常重要的技术。本实例在页面中设置一个访问权限，判断用户是否可以访问该网页。如果用户名和密码不正确就不可以访问，并弹出提示对话框重定向到该网页，运行结果如图 3.64 所示。



图 3.64　弹出提示对话框并重定向网页

关键技术

本实例主要将 PHP 与 JavaScript 相结合，通过 JavaScript 脚本来弹出对话框信息，应用 window 对象中的 alert() 方法和窗口属性 location 实现此功能，语法格式如下：

```
alert ('弹出对话框的内容');
window.location.href='重新定向的页面';
```

参数说明：

alert()：弹出一个警告对话框。

location：定位区，也叫地址栏，是可以输入 URL 的浏览器文本区。

设计过程

弹出提示对话框并重定向网页文件的关键代码如下：

```
<?php session_start();?>
<title>弹出提示对话框并重定向网页</title>
<?php
if($gly==true && $mima==true){
if($ljid==true){
$ljid=$_GET['ljid'];
echo "这就是弹出提示对话框并重定向网页!";}
?>
<?php }else{
echo "<script>alert ('这就是弹出提示对话框并重定向网页!'); window.location.href='admin.php';</script>";
}
?>
```

秘笈心法

心法领悟 208：浅谈网页重定向。

这里的网页重定向与 PHP 判断用户登录权限页相似，都是网页的一种跳转方法。

实例 209	自动关闭的广告窗口 光盘位置：光盘\MR\03\209	高级 趣味指数：★★★★

实例说明

进入网站后，很多网站都会弹出一个新窗口（如广告等），多数窗口需要浏览者自行关闭。为了方便浏览者对页面中信息的浏览，本实例运用一种新的方法来解决弹出窗口关闭这一问题。当浏览者进入某网站进行浏览时，无需关闭弹出的新窗口，在页面运行超过一定的时间之后，该窗口将自动关闭，这将大大方便浏览者的浏览。运行本实例，将弹出一个新窗口显示广告信息，如图 3.65 所示，该广告窗口在页面运行 5 秒钟后自动关闭。



图 3.65　自动关闭的广告窗口

关键技术

本实例主要应用 window 对象的 setTimeout()方法，实现延迟执行某一操作。window 对象的 setTimeout()方法的语法格式如下：

```
setTimeout(expression,secdelay[,language])
```

参数说明：

expression：是一个字符串，可包含任何对函数、方法或者单个 JavaScript 语句的调用。

secdelay：指定运行间隔的时间，以毫秒为单位。

language：指定语句或参数 expression 调用的函数所使用的脚本语言。如果完全使用 JavaScript，则不需设置该参数。

设计过程

（1）通过 JavaScript 脚本实现打开新窗口功能，代码如下：

```
<script language="javascript">
window.open("new.htm","new","height=135,width=180,top=10,left=20");
</script>
```

（2）通过设置 window 对象的 setTimeout()方法，实现弹出新窗口的自动关闭，代码如下：

```
<body onLoad="window.setTimeout('window.close()',5000)">
```

秘笈心法

心法领悟 209：秒与毫秒。

如果用户不是运动员，很少能接触到毫秒这个单位。但是在计算机领域中，毫秒是经常被用到的。秒与毫秒的换算比例是 1:1000，也就是说 1 秒等于 1000 毫秒。

实例 210　在弹出的对话框中选择个性头像

光盘位置：光盘\MR\03\210

高级

趣味指数：★★★★

实例说明

在论坛、博客等注册页面添加头像选择功能，可以增加网页活力。运行本实例，首先单击图 3.66 中的“头像选择”超链接，在弹出的窗口中选择自己喜爱的头像后，弹出窗口将自动关闭，注册页面中的头像变成用户所选择的头像。

图 3.66　选择头像

关键技术

本实例通过调用 window 对象的 showModalDialog()方法弹出一个窗口，当用户单击某个头像后，将该头像对应的数值传递给父窗口，父窗口通过传递过来的数值指定用户所选择的头像。

设计过程

（1）首先在窗口中添加“头像选择”超链接，以便调用 select_face()函数实现窗口的弹出功能，代码如下：

```
[<a href="#" onclick="select_face()" class="a1">头像选择</a>]
```

（2）利用 Javascript 编写 select_face()函数，实现头像选择窗口的弹出以及返回用户所选择头像对应的数值，代码如下：

```
<script language="javascript">
function select_face(){
var somevalue;
somevalue=window.showModalDialog("browse.php","请选择您喜欢的人物头像", "dialogWidth='310px';dialogHeight='240px';status='no';help='no';
scrollbars='no'");
document.form_regist.user_src.src="images/head/"+somevalue+".gif";
 }
</script>
```

（3）在弹出的窗口中通过 selectface()函数返回用户所选择头像对应的数值，代码如下：

```
<script language="javascript">
function selectface(ICQ){
window.returnValue=ICQ;
window.close();
}
</script>
```

秘笈心法

心法领悟 210：showModalDialog()方法详解。

```
Window.showModalDialog('弹出窗口地址', '弹出窗口名称', '弹出窗口所具有的属性')
```

窗口所具有的属性包括窗口的高度、宽度，以及窗口是否有地址栏、状态栏、滚动条等一些特性。

实例 211	浮动的广告窗口 光盘位置：光盘\MR\03\211	高级 趣味指数：★★★★

实例说明

网络不仅是信息的海洋，也是广告的海洋。除了普通的 gif 和 flash 之外，浮动广告也是时下网上较为流行的广告形式之一，当拖曳浏览器的滚动条时，这种页面上浮动的广告，可以跟随屏幕一起移动，这种效果对于广告展示有相当大的实用价值。本实例将制作一个浮动的广告窗口，运行结果如图 3.67 所示。

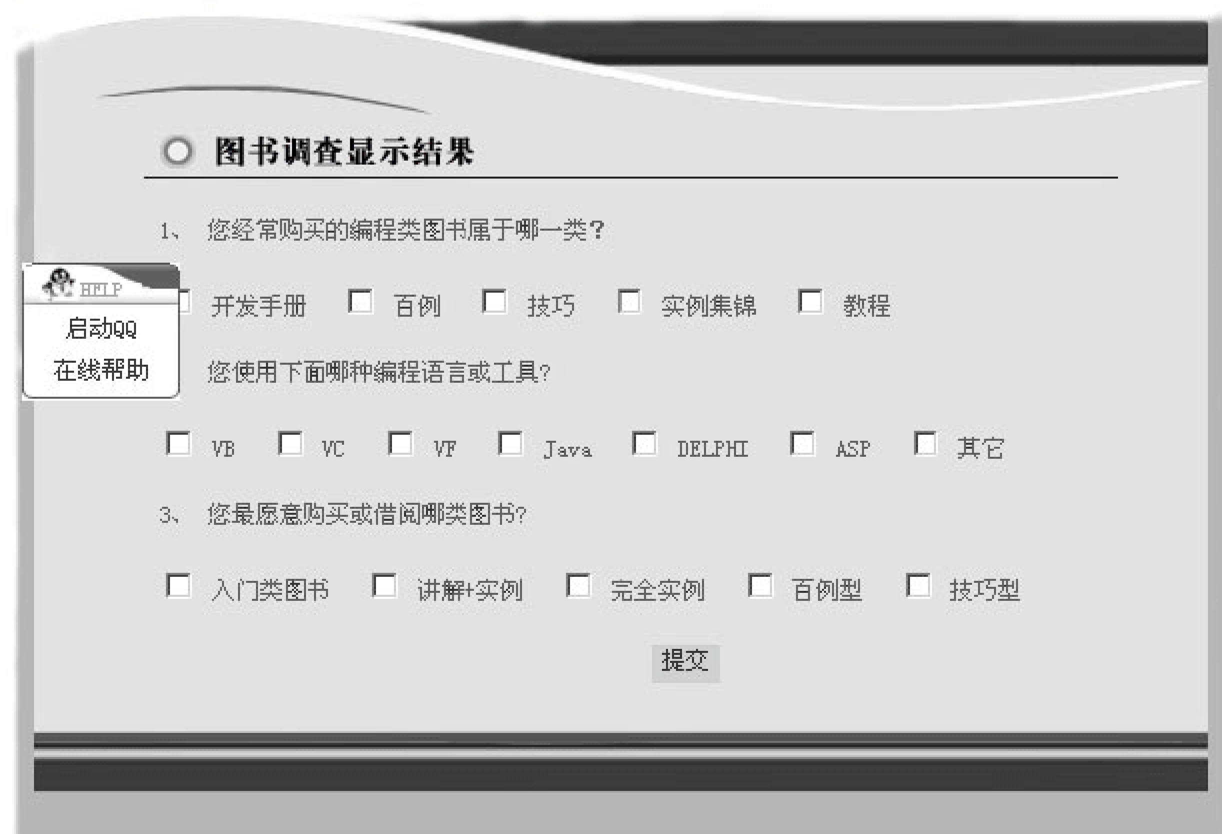


图 3.67　浮动的广告窗口

关键技术

本实例主要应用 DIV 层制作浮动广告，然后应用 document.write()方法输出 DIV 层。document.write()方法的语法格式如下：

```
document.write(value,…)
```

参数说明：

value：要添加到 document 的任意一个 Javascript 值。如果该值不是字符串，那么在被添加文档之前，会被转化成一个字符串。

…：任意多个（0 个或多个）要（依次）写入文档的值。

设计过程

（1）在网站首页引入 JavaScript 脚本文件，代码如下：

```
<SCRIPT language=JavaScript src="js/left.js"></SCRIPT>
```

（2）运用 JavaScript 脚本自定义函数控制浮动广告的位置及其相关属性，代码如下：

```
var adve=0.15
var collection;
function motion() {
 this.items= [];
 this.addItem= function(id,x,y,content){
 document.write('<DIV id='+id+' style="Z-INDEX: 10; POSITION: absolute;    width:80px;
height:60px;left:'+(typeof(x)=='string'?eval(x):x)+';top:'+(typeof(y)=='string'?eval(y):y)+'">'+content+'</DIV>');
 var newItem= {};
 newItem.object= document.getElementById(id);
 newItem.x= x;
 newItem.y= y;
 this.items[this.items.length]= newItem;
   }
 this.play=function(){
 collection=this.items
 setInterval('play()',10);
   }
 }
 function play()        {
 for(var i=0;i<collection.length;i++)          {
 var followOb= collection[i].object;
 var followObj_x       = (typeof(collection[i].x)=='string'?eval(collection[i].x):collection[i].x);
 var followObj_y= (typeof(collection[i].y)=='string'?eval(collection[i].y):collection[i].y);
 if(followObj.offsetLeft!=(document.body.scrollLeft+followObj_x)) {
 var dx=(document.body.scrollLeft+followObj_x-followObj.offsetLeft)*adve;
 dx=(dx>0?1:-1)*Math.ceil(Math.abs(dx));
 followObj.style.left=followObj.offsetLeft+dx;
 }
if(followObj.offsetTop!=(document.body.scrollTop+followObj_y)) {
var dy=(document.body.scrollTop+followObj_y-followObj.offsetTop)*adve;
dy=(dy>0?1:-1)*Math.ceil(Math.abs(dy));
 followObj.style.top=followObj.offsetTop+dy;
 }
 followObj.style.display      = '';
 }
 }
 var themotion= new motion();
 themotion.addItem('followDiv1',200,200,'<a href="http://wpa.qq.com/msgrd?V=1&Uin=228691050&Site=在线咨询&Menu=no" title="在线即时交谈
"><img src=./images/qqhelp.gif    height=70 width="80" border=0></a>');
 themotion.play();
```

秘笈心法

心法领悟 211：使用 document.write()方法的注意事项。

调用方法 document.write()的结果也许不能立刻在目标浏览器窗口中显示出来，这是因为浏览器可能会将数据缓存起来，以便形成大的数据块输出。

实例 212 在页面右下角弹出渐显的广告窗口

光盘位置：光盘\MR\03\212

高级

趣味指数：★★★★

实例说明

广告作为网站最大的盈利手段，任何网站都不可能将其省略，那么，如何才能实现网站既可以从广告中获利，又能使用户流畅地阅读网站信息呢？本实例将介绍如何制作一个从首页右下角弹出的渐显广告窗体，该窗体在用户登录网站时自动从右下角渐渐弹出，然后由用户手动关闭。本实例的运行结果如图 3.68 所示。



图 3.68　右下角渐显广告

关键技术

本实例主要应用 JavaScript 脚本的 window 对象，window 对象主要用于在 HTML 中打开窗口，应用极为普遍，但也有一些缺陷。用户浏览器决定窗口的外观，设计者左右不了其窗口的大小及样式，但 JavaScript 给了程序这种控制权。在 JavaScript 中，可以使用 window 对象来实现对窗口的控制。

设计过程

本实例主要应用 window 对象的 open()方法打开一个在首页右下角弹出渐显的广告窗口，在网站首页中添加如下代码：

```
<script language="javascript">
var newformW=300;
var newformH=180;
function pp(){
 var T=window.screen.width-newformW;
 var L=window.screen.height+newformH;
 var name=window.open("advertise.htm","","width="+newformW+",height="+newformH+",top="+T+",left="+L);
}
pp();
</script>
```

秘笈心法

心法领悟 212：window 对象打开窗口详解。

使用 window 对象打开窗口的语法格式如下：

```
windowVar=window.open(url,windowname[,location]);
```

参数说明：

windowVar：当前打开窗口的句柄。如果 open()方法成功，则 windowVar 的值为一个 window 对象的句柄，

否则 windowVar 的值是一个空值。

url：目标窗口的 URL。如果 URL 是一个空字符串，则浏览器将打开一个空白窗口，允许用 write()方法创建动态 HTML。

windowname：window 对象的名称。

location：对窗口属性进行设置。

3.7　状态栏设计

不论是在应用程序中还是在网页中都会应用到状态栏，状态栏可为用户提供一些提示信息，如当前系统的日期和时间、软件版本、链接地址、系统信息以及一些自定义信息等。

实例 213	跑马灯文字 光盘位置：光盘\MR\03\213	高级 趣味指数：★★★★

实例说明

浏览网页时，用户往往更加注意网站页面的内容，而不会过多注意状态栏，如果为页面的状态栏添加一些特效，一定会给网站增添一道亮丽的风景。本实例实现在状态栏中显示跑马灯特效文字，运行结果如图 3.69 所示。



图 3.69　跑马灯文字

关键技术

本实例主要应用 JavaScript 脚本的 window 对象的 status 属性，并通过编写自定义函数来设置状态栏中的文字及文字属性，通过 if...else 条件语句来控制文字的循环显示效果。

最后，在<body>标记中调用该函数，从而实现状态栏中的跑马灯文字特效。

设计过程

（1）利用 JavaScript 脚本编写自定义函数，生成状态栏中的文字及文字属性，代码如下：

```
<SCRI PT language="JavaScript">
function scrollit(s){
var text= "        您好！欢迎访问明日科技图书网站。客服热线：0431-84978981"
var printtext = " ";
var i = 1;
if (s > 210){
```

```
s--;
var out="scrollit("+s+")";
var timerTwo=window.setTimeout(out,200);
}
else if (s <= 210 && s > 0){
for (i=0;i<s;i++){
printtext+=" ";
}
printtext+=text;
s--;
window.status=printtext;
out="scrollit("+s+")";
timerTwo=window.setTimeout(out,200);
}
else if (s <= 0){
if (-s < text.length){
printtext+=text.substring(-s,text.length);
s--;
window.status=printtext;
out="scrollit("+s+")";
timerTwo=window.setTimeout(out,200);
}
else{
window.status=" ";
timerTwo=window.setTimeout("scrollit(100)",200);
}
}
}
</SCRIPT>
```

（2）在<body>标记中调用自定义的函数。

```
<body onLoad="scrollit(100);">
```

秘笈心法

心法领悟 213：window 对象的 status 属性详解。

window 对象的 status 属性的语法格式如下：

```
window.status
```

属性 status 是一个可读可写的字符串，声明浏览器状态栏的当前内容。

实例 214	状态栏中的收缩显示文字 光盘位置：光盘\MR\03\214	高级 趣味指数：★★★★

实例说明

为了在节省网页空间的同时更好地服务于广大的用户，可以将网站要显示的简短的广告信息显示在状态栏中。运行本实例，在状态栏中收缩显示明日科技公司的网址和客服热线，运行结果如图 3.70 所示。



图 3.70　收缩显示文字

关键技术

本实例主要应用 window 对象的 status 属性。当显示的文字总长度等于该文本的长度时，再反方向滚动文字，从而实现在状态栏中收缩显示文字。

window 对象的 status 属性的语法格式如下：

```
window.status
```

属性 status 是一个可读可写的字符串，声明了浏览器状态栏的当前内容。

设计过程

本实例主要通过 JavaScript 脚本编写状态栏中收缩显示文字特效的自定义函数，完整代码如下：

```
<Script language=JavaScript>
var text1="欢迎访问明日科技主页：www.mingrisoft.com";
comeback=0;
cometo=0;
function dis(){
window.status=text1.substring(0,cometo);
if(comeback==1){
 cometo--;
 if(cometo==0){comeback=0}
}
else {cometo++;if(cometo==text1.length){comeback=1}}
window.status=text1.substring(0,cometo)+"(客服热线：0431-84978981)";
if(cometo==text1.length){window.setTimeout("dis()",1000)}
else {window.setTimeout("dis()",50)}
}
dis();
</Script>
```

秘笈心法

心法领悟 214：状态栏。

状态栏其实只是起到一个类似于广告的作用，只不过它比其他的弹出层广告要逊色一些。

实例 215	状态栏中显示数字时钟 光盘位置：光盘\MR\03\215	高级 趣味指数：★★★★

实例说明

在状态栏中显示数字时钟是个性化网站所特有的功能，它不仅可以节约网页空间，而且可以方便用户时时查看。本实例实现在状态栏中显示一个 24 小时制的数字时钟，运行结果如图 3.71 所示。



图 3.71　状态栏显示时钟

关键技术

本实例主要应用 window 对象的 status 属性实现在状态栏中显示数字时钟，具体详细讲解请参考实例 214。

设计过程

（1）本实例主要通过 JavaScript 脚本开发状态栏中显示动态数字时钟的自定义函数，代码如下：

```
<script language="JavaScript">
var timerID = null;
var timerRunning = false;

function stopclock (){
	if(timerRunning)
		clearTimeout(timerID);
		timerRunning = false;
}

function showtime () {
	var now = new Date();
	var hours = now.getHours();
	var minutes = now.getMinutes();
	var seconds = now.getSeconds()
	var timeValue = "北京时间： " + ((hours >24) ? hours -24 :hours)
	timeValue += ((minutes < 10) ? ":0" : ":") + minutes
	timeValue += ((seconds < 10) ? ":0" : ":") + seconds
	window.status = timeValue;
	timerID = setTimeout("showtime()",1000);
	timerRunning = true;
}

function startclock () {
	stopclock();
	showtime();
}
</script>
```

（2）在<body>标记中调用自定义函数 startclock()，代码如下：

```
<body onLoad="startclock()">
```

秘笈心法

心法领悟 215：JavaScript 脚本中时间的设置。

在 JavaScript 脚本中设置日期和时间，首先要实例化 Date 类，然后调用其中的方法，获取对应的日期、时间值。

3.8　导航菜单设计

网站导航菜单对于每个网站都是必不可少的，它相当于城市中的路标，可以指引浏览者找到自己热衷的内容。导航菜单有很多种，如本节中介绍的带图标的文字导航条、flash 导航条、按钮导航条、弹出式下拉菜单、二级导航菜单、树状菜单等。

实例 216	树状导航菜单 光盘位置：光盘\MR\03\216	高级 趣味指数：★★★★

实例说明

树形结构能够以层次形式展示信息，用它来描述具有上下级关系的内容再恰当不过。本实例利用菜单的树

形结构来描述企业的人事组织架构，效果如图 3.72 所示。



图 3.72　树形导航菜单

关键技术

本实例中实现菜单的树形显示是通过<table>标签结合 DIV 实现的。父节点利用表格来显示，子节点利用 DIV 嵌套表格来实现。这样做的目的是容易对子节点的显示和隐藏进行控制。例如，当用户单击父节点时，如果子节点没有展开，将 DIV 设置为可见就可以显示子节点；如果子节点可见，将 DIV 设置为不可见就可以隐藏子节点。

设计过程

（1）使用 Dreamweaver 创建一个 PHP 动态页，保存为 index.php 文件。

（2）创建一个表单，在表单中添加一个 2 行 1 列的表格，在表格的第 2 行再添加一个 2 行 2 列的表格。表格的第 1 列用于显示节点前的图标，第 2 列描述节点文本，主要代码如下：

```
<form id="form1" name="form1" method="post" action="" >
  <table width="372"  border="1" align="center"  bgcolor="#E6F2F2" bordercolor="#478D8D"
  style="border-style:none" cellspacing="0" >
    <tr>
      <td    scope="col" align="center" style="border-bottom-style:none" >人事组织架构</td>
    </tr>
    <tr>
      <td >
           <table width="100%"  border="0" cellpadding="0" cellspacing="0" align="left">
        <?php                                                            //嵌入 PHP 脚本

          $xml_file = simplexml_load_file("org.xml");                    //从 XML 文件中加载节点
              foreach($xml_file->children() as $node){                   //遍历根节点
      ?>
          <tr>
            <td width="38%" align="right">                               //设置节点图标
                  <img src="images/rootnode.gif" width="15" height="15" border="0" >   </img>
            </td>
             <td   width ="62%"   left="0px" align="left" ><a href="#" onclick= "ShowNode(<?php echo $node ?>)">  
            <?php echo $node ?></a>                                      //设置节点文本
             </td>
          </tr>
          <tr>
          <td width ="38%" align="center" >
          </td>
             <td   width ="62%" >
                   <div id= <?php echo $node ?> style = "display:none" class="HideDiv" >  //定义 DIV
                     <table width="100%" border="0" cellpadding="0" cellspacing="0" align="left" >
                         <?php
                           foreach($node->children() as $value){                //遍历子节点
                         ?>
                         <tr>
                       <td width ="10%"   align="right" >
                               <img   src="images/childnode.gif" width="15" height="15" border="0" > </img>
                         </td>
                         <td   width ="90%"   align="left"   ><a href="#" >  <?php echo $value?></a>
                         </td>
                         </tr>
                   <?php
```

```
                                            }
                                        ?>
                                    </table>
                                </div>
                            </td>
                        </tr>
                    <?php
                    }
                        ?>
                    </table>
                </td>
            </tr>
        </table>
        <script language="javascript">
        function ShowNode(ParentID) {                                    //隐藏或显示子节点
            var Div = ParentID;
            if (Div.style.display == "none"){
                Div.style.display = "block";                             //显示 DIV，以此来显示子节点
            }else if (Div.style.display == "block"){
                Div.style.display = "none";                              //隐藏 DIV，间接隐藏子节点
            }
        }
        </script>
</form>
```

（3）人事组织的架构存储于 XML 文件 org.xml 中。

秘笈心法

心法领悟 216：动态设置超链接。

在设计网站项目时，有时需要根据实际情况动态设置超链接的内容。在利用标签<a>定义超链接时可以结合 PHP 脚本来实现这一功能，例如：

```
<a href="#" onclick= "ShowNode(<?php echo $node ?>)">   <?php echo $node ?></a>
```

实例 217　收缩式导航菜单

光盘位置：光盘\MR\03\217

高级

趣味指数：★★★★

实例说明

在网站中不仅可以设置导航条，而且还可以设置导航菜单。由于菜单内容比较多，在同一页面中显示会比较杂乱，所以目前大多设计者都采用收缩式的导航菜单。运行本实例，当浏览者单击“网站管理”超链接时，在其下方将弹出导航菜单，如图 3.73 所示，浏览者再次单击“网站管理”超链接时，导航菜单又收缩回去，页面中不再显示菜单中的内容。



图 3.73　收缩式菜单

关键技术

本实例主要是利用显示、隐藏表格来实现收缩式导航菜单的功能。单击导航超链接，显示当前菜单的内容，隐藏上一个显示的菜单。在隐藏菜单时，让其有规律地隐藏进而实现动画效果。

设计过程

（1）显示菜单的自定义函数，代码如下：

```
<SCRIPT language=javascript>
function show(obj,maxg,obj2){
   if(obj.style.pixelHeight<maxg) {
      obj.style.pixelHeight+=maxg/10;
 obj.filters.alpha.opacity+=20;
 obj2.background="images/title_hide.gif";
      if(obj.style.pixelHeight==maxg/10)
   obj.style.display='block';
 myObj=obj;
 mymaxg=maxg;
 myObj2=obj2;
 setTimeout('show(myObj,mymaxg,myObj2)','5');
   }
}
</SCRIPT>
```

（2）隐藏菜单的自定义函数，代码如下：

```
<SCRIPT language=javascript>
function hide(obj,maxg,obj2){
   if(obj.style.pixelHeight>0) {
      if(obj.style.pixelHeight==maxg/5)
   obj.style.display='none';
      obj.style.pixelHeight-=maxg/5;
 obj.filters.alpha.opacity-=10;
 obj2.background="images/title_show.gif";
 myObj=obj;
 mymaxg=maxg
 myObj2=obj2;
 setTimeout('hide(myObj,mymaxg,myObj2)','5');
   }
   else
      if(whichContinue)
   whichContinue.click();
}
</SCRIPT>
```

（3）单击菜单上的文字超链接时，隐藏前一个菜单，显示当前菜单，代码如下：

```
<SCRIPT language=javascript>
function chang(obj,maxg,obj2){
   if(obj.style.pixelHeight) {
      hide(obj,maxg,obj2);
      nopen='';
      whichcontinue='';
   }
   else
      if(nopen){
         whichContinue=obj2;
         nopen.click();
 }
 else{
    show(obj,maxg,obj2);
    nopen=obj2;
    whichContinue='';
 }
}
</SCRIPT>
```

（4）在表格的相关鼠标事件中调用自定义的方法和属性来改变收缩菜单的显示和隐藏，代码如下：

```
<TD class=list_title id=list1  onmouseover="this.typename='list_title2';"  onclick=chang(menu1,60,list1);  onmouseout="this.typename='list_title';"
background="images/title_hide.gif"  height=25><SPAN>网站管理</SPAN> </TD>
```

秘笈心法

心法领悟 217：收缩式导航菜单。

收缩式导航菜单是笔者最喜欢的导航菜单，虽然只是由一些 JavaScript 事件组成，但是操作时的动态效果很强。

实例 218	展开式导航菜单 光盘位置：光盘\MR\03\218	高级 趣味指数：★★★★

实例说明

对于一个企业网站来说，“联系我们”超链接是必不可少的。为了更加引人注意，可以将其制作成具有展开式动画效果的导航条。运行本实例，可以看到页面右侧的“联系我们”导航条是逐渐展开的，并且每次刷新页面时，都会以同样的动画效果展开，展开效果如图 3.74 所示。



图 3.74　展开式导航条

关键技术

本实例主要是通过自定义 JavaScript 函数控制图片（image）对象的 height 属性值实现的。为了实现动画效果，在 JavaScript 函数中还需要应用 setTimeout()方法延迟执行改变图片高度的操作。

设计过程

（1）在网页中显示“联系我们”导航条的位置插入一张全部展开后的“联系我们”导航图片，在需要设置超链接的文字上添加图片热点并设置相应的超链接。

（2）将图片的 height 属性值设置为 0，name 属性设置为 our。

（3）编写自定义的 JavaScript 函数 ourmove()，用于实现图片的展开效果。ourmove()函数的代码如下：

```
<script language="javascript">
function ourmove(){
 if(our.height<163){
        our.height=our.height+3
        setTimeout(ourmove,1)
 }
}
</script>
```

（4）在当前页面的最底部，也就是</body>标记之前，添加如下代码实现导航条的展开式动画效果。

```
<script language="javascript">
 ourmove();
</script>
```

秘笈心法

心法领悟 218：使用 ourmove()函数的注意事项。

一定要将调用 ourmove()函数的代码放在</body>标记之前，否则页面的其他内容就要等到动画显示完毕才可以显示，达不到预期的效果。

实例 219　解释型菜单

光盘位置：光盘\MR\03\219

高级

趣味指数：★★★★

实例说明

在设计网站的导航菜单时，有时为了让用户快速了解菜单导航的详细内容，需要为菜单设置提示信息。这样当用户鼠标移动到菜单项上时，就会显示提示信息，效果如图 3.75 所示。

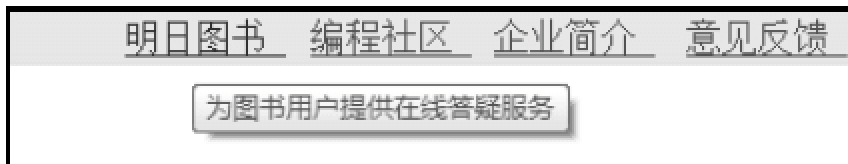


图 3.75　解释型菜单

关键技术

本实例主要应用超链接中的 title 属性实现解释型菜单。在定义超链接时为其设置 title 属性，就可以实现菜单的提示功能，例如：

```
<a href="#" title = "为图书用户提供在线答疑服务"> 明日图书   </a>
```

设计过程

（1）使用 Dreamweaver 创建一个 PHP 动态页，保存为 index.php 文件。

（2）创建一个表单，在表单中添加一个 3 行 1 列的表格，在表格的第 2 行再添加一个 1 行 4 列的表格。

（3）在嵌套的表格中加入超链接，主要代码如下：

```
<td width="123" scope="col" align="right"> <a href="#" title = "为图书用户提供在线答疑服务"> 明日图书   </a></td>
<td width="79" scope="col" align="right"><a href="#" title = "为编程人员提供一个在线交流平台">编程社区  </a></td>
<td width="78" scope="col" align="right"><a href="#">企业简介  </a></td>
<td width="82" scope="col" align="right"><a href="#">意见反馈  </a></td>
```

秘笈心法

心法领悟 219：使用 CSS 样式统一超链接效果。

在设计表单页面时，通常需要统一某一类表单元素的效果。例如，统一超链接文本字体的大小、文本颜色等。一种简单的方式是利用 CSS 样式来实现，例如：

```
<style>
a{
font-size:16px;
color:#900;
}
```

这样页面中的超链接字体均为 16 个像素，颜色为暗红色。

实例 220　半透明背景的下拉菜单

光盘位置：光盘\MR\03\220

高级

趣味指数：★★★★

实例说明

网站中的导航菜单是网站首页必不可少的表单元素。美观、新颖的导航菜单会为网站增色许多。本实例将实现一个半透明效果的菜单，效果如图 3.76 所示。



图 3.76　半透明效果菜单

关键技术

半透明效果的菜单可以通过 CSS 样式中的滤镜 filter 来实现。在滤镜中使用 alpha 属性设置透明程度（范围 0~100，0 表示完全透明，100 表示不透明），其语法格式如下：

```
{Filter:alpha(opacity=opacity,finishopacity=finishopacity,sytle=sytle,startx=startx,starty=starty,finishx=finishx,
finishx=finishyx)}
```

例如，下面的语句将滤镜透明程度设置为 50%。

```
Filter: Alpha(Opacity=50);
```

设计过程

（1）使用 Dreamweaver 创建一个 PHP 动态页，保存为 index.php 文件。

（2）创建一个表单，在表单中添加一个 3 行 1 列的表格，在第 2 行中再添加一个 1 行 3 列的表格。

（3）定义 CSS 样式表，设置超链接的效果和半透明菜单的效果，代码如下：

```
<style type="text/css">
A {
 COLOR: blue; FONT-SIZE: 10pt; TEXT-DECORATION: none;color:#FF6600
}
A:hover {
 COLOR: black; FONT-SIZE: 10pt;color:#395531
}
.transMenu {
	BORDER: #666666 1px solid;
	FONT: 12px Verdana;
	POSITION: absolute;
	background-image:url("images/item_out.jpg");
	background-repeat : repeat-y;
	Filter: Alpha(Opacity=50);
}
</style>
```

（4）利用 JavaScript 脚本控制菜单的显示与隐藏，代码如下：

```
<script language="javascript">
 function ShowMenu(divID, menuItem){                              //显示子菜单
	if (document.all){
		switch (menuItem){
			case '网站导航':
			{
				var htmltxt = "<a href = 'www.mingribook.com'> 明日图书网站 </a> <br> <a href =
 'www.mrbccd.cn'> 明日编程社区 </a> <br> <a href = 'www.mrsoft.com'> 明日软件 </a> <br>";
				MenuID.innerHTML = htmltxt;             //设置 DIV 代码
				MenuID.style.padding = "5px";           //设置 DIV 边距
				MenuID.style.height = "55px";           //设置 DIV 高度
				break;
			}
			case '技术支持':
			{
				var menutxt = "<a href = 'www.mingribook.com'> QQ </a> <br> <a href = 'www.mrbccd.cn'>
 MSN </a> <br> <a href = 'www.mrsoft.com'> Phone </a> <br>";
				MenuID.innerHTML = menutxt;             //设置 DIV 代码
				MenuID.style.padding = "5px";           //设置 DIV 边距
				MenuID.style.height = "55px";           //设置 DIV 高度
```

```
                        break;
                    }
                }
                //设置菜单显示位置
                MenuID.style.left = document.body.scrollLeft + event.clientX - event.offsetX;
                MenuID.style.top = document.body.scrollTop + event.clientY - event.offsetY + 18 ;
                MenuID.style.display = "block";
        }
}
function HideMenu(){                                          //隐藏子菜单
        if (document.all){
                MenuID.style.display = "none";                //隐藏 DIV
        }
}
if (document.all){
        document.onmousemove = HideMenu;                      //鼠标在表单中移动时隐藏菜单
}
</script>
```

秘笈心法

心法领悟 220：设置鼠标移动到超链接的效果。

在使用超链接时通常会为超链接设置一些特效。例如，当用户鼠标移动或单击超链接时文本颜色发生变化。要实现这一功能最简单的方式是定义 CSS 样式表，例如：

```
A:hover {
FONT-SIZE: 10pt;
color:#395531
}
```

实例 221　二级导航菜单

光盘位置：光盘\MR\03\221

高级

趣味指数：★★★★

实例说明

二级导航菜单是许多网站通常采用的菜单导航形式。顶行显示一级菜单，其下显示二级菜单。当用户选择不同的一级菜单时，二级菜单会动态发生变化。本实例就实现这种效果的导航菜单，如图 3.77 所示。



图 3.77　二级导航菜单

关键技术

实现二级导航菜单的关键是二级菜单的显示。当用户选择不同的顶层菜单时需要显示不同的二级菜单。这里我们用 DIV 作为二级导航菜单的容器，通过动态设置 DIV 对象的 innerHTML 属性来实现动态显示二级菜单的效果。

设计过程

（1）使用 Dreamweaver 创建一个 PHP 动态页，保存为 index.php 文件。

（2）创建一个表单，在表单中添加一个 2 行 1 列的表格，在第 1 行中再添加一个 1 行 4 列的表格，在第 2 行再添加一个 1 行 2 列的表格。表单代码如下：

```
<form id="form1" name="form1" method="post" action="">
  <table width="610" height="441" border="0" align="center">
    <tr>
```

```
        <th height="62" align="right" valign="bottom" scope="col"><table width="337" border="1" align="right" cellpadding="0" cellspacing="0"
bordercolor="#993399" background="images/menuBK.JPG">
            <tr>
                <td width="327" scope="col"> <a href ='#' onmousemove= MenuCtrl('论坛中心','subMenu');>论坛中心</a></td>
                <td width="327" scope="col"><a href = '#' onmousemove=MenuCtrl('勘误发布','subMenu');>勘误发布</a></td>
                <td width="327" scope="col"><a href = '#' onmousemove=MenuCtrl('作品展示','subMenu');>作品展示</a></td>
                <td width="327" scope="col"><a href = '#' onmousemove=MenuCtrl('联系我们','subMenu');>联系我们</a></td>
            </tr>
        </table></th>
    </tr>
    <tr valign="top">
        <td height="373"><table width="606" border="0" align="center">
            <tr>
                <th width="283" scope="col"> </th>
                <th width="313" scope="col"><div id="subMenu" align="left"></div></th>
            </tr>
        </table></td>
    </tr>
  </table>
</form>
```

（3）利用 JavaScript 脚本控制二级菜单的显示，代码如下：

```
<script language="javascript">
 var curSel = "";
 function   MenuCtrl(menuID, divID){
        var divObj = document.getElementById(divID);                              //获取 DIV 对象
        if (divObj == null){
                return;
        }
        switch (menuID){                                                          //判断顶层菜单
                case '论坛中心':
                {
                        //在 DIV 中显示子菜单
                        divObj.innerHTML = "<a href = '#'>Visual C++ |<a href = '#'> Visual Basic |<a href = '#'> PHP </a>";
                        curSel ='论坛中心';
                        break;
                }
                case '勘误发布':
                {
                        //在 DIV 中显示子菜单
                        divObj.innerHTML = "<a href = '#'>光盘错误  |<a href = '#'>  文档错误  |<a href = '#'>目录错误  </a>";
                        curSel = '勘误发布';
                        break;
                }
                case '作品展示':
                {
                        //在 DIV 中显示子菜单
                        divObj.innerHTML = "<a href = '#'>视频学 VB |<a href = '#'>  视频学 VC |<a href = '#'>
视频学 Asp.net |<a href='#'>  视频学 C# </a>";
                        curSel = '作品展心';
                        break;
                }
                case '联系我们':
                {
                        //在 DIV 中显示子菜单
divObj.innerHTML = "<a href = '#'>OICQ |<a href = '#'> MSN|<a href = '#'> Phone</a>";
                        curSel ='联系我们';
                        break;
                }
        }
 }
 if (document.all){
        MenuCtrl('论坛中心', 'subMenu');                                          //默认情况下显示第一个顶层菜单的二级菜单项
 }
</script>
```

秘笈心法

心法领悟 221：设置表格单边框效果。

默认情况下，表格具有双边框，在利用表格进行界面布局时这种效果会影响页面的美观性。我们可以通过将表格的 cellspacing 属性设置为 0 来实现表格的单边框效果，例如：

```
<table width="337" border="1"   cellspacing="0" bordercolor="#993399" background="images/menuBK.JPG">
```

实例 222	导航条的动画效果 光盘位置：光盘\MR\03\222	高级 趣味指数：★★★★

实例说明

导航条是网站设计中不可缺少的元素之一，它能正确地引导浏览者查找需要的资料，成为浏览者的网站路标。同时网页导航条的设计风格也能影响到页面的整体风格，对于一个静态元素居多的页面，可以为导航条添加动态效果，这样可以使整个网站不至于太呆板，增强网页的欣赏价值。例如，在明日实业网站中，笔者将导航条设计成动画效果，用户将鼠标移动到任意一个导航按钮上时，该按钮都会突出显示，鼠标移出后，又恢复为原来的位置，运行结果如图 3.78 所示。

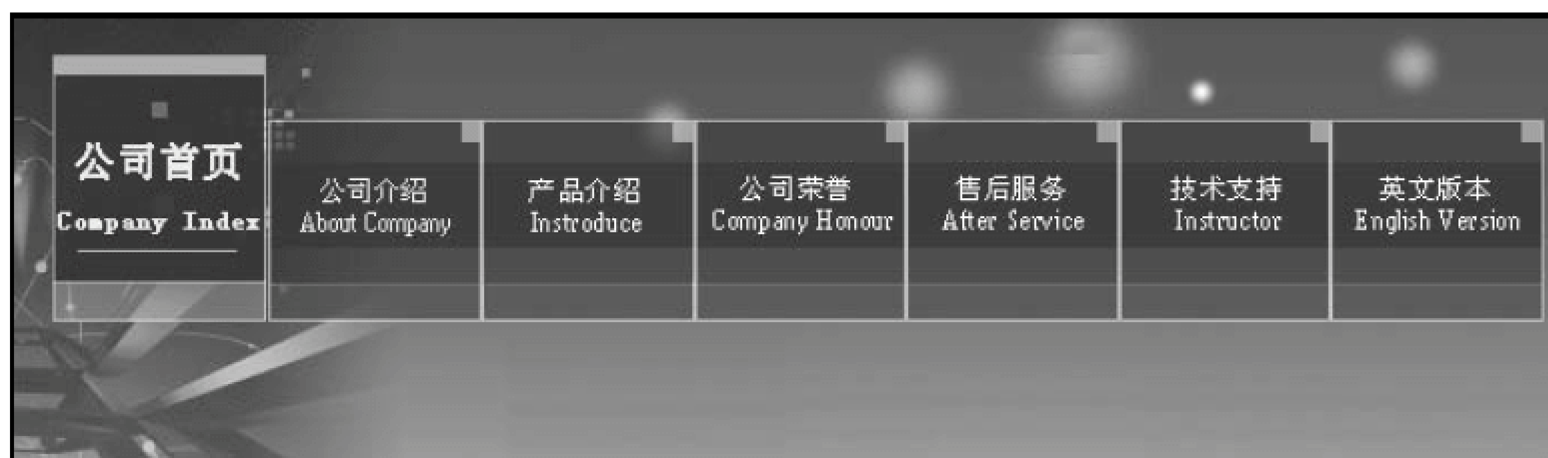


图 3.78　导航条的动画效果

关键技术

本实例主要是通过 Image 对象的鼠标事件控制 Image 对象的 src 属性的值实现的。

在 JavaScript 里提供了对图像进行处理的专用对象 Image 来装入文档的图形。Image 对象与其他对象的差别在于，其允许通过构造器显示和创建新的 Image 对象，创造和预装入的图形之前并非 Web 页面的组成部分。Image 对象存在于浏览器的缓冲区中，用于替换已经显示的图像。

用 Image()构造器创建图形的语法格式如下：

```
ObjImg=new image()                          //创建了一个新的 Image 对象，并将其赋予变量 ObjImg
ObjImg.src="图片文件相对路径"                //设置 Image 对象的 Src 属性
```

设计过程

（1）准备 14 张图片。7 张鼠标移出时显示的图片，图片的名称为“menu_0+1~7 的数字+.gif”，7 张鼠标移入时显示的图片，图片的名称为“menu_0+1~7 的数字+_over.gif”。

（2）将准备的 7 张鼠标移出时显示的图片按顺序插入到页面中的适当位置，并设置其鼠标事件 onMouseMove 和 onMouseout 执行的操作，这里分别调用两个不同的自定义 JavaScript 函数，关键代码如下：

```
<img src="Images/top/menu_01.gif" id="image1" width="95" height="119" border="0" onMouseMove="move(this,'1')" onMouseout="out(this,'1')">
<img src="Images/top/menu_02.gif" id="image2" width="95" height="119" border="0" onMouseMove="move(this,'2')" onMouseout="out(this,'2')">
<img src="Images/top/menu_03.gif" name="image3"   width="95" height="119" border="0" onMouseMove="move(this,'3')"
onMouseout="out(this,'3')">
<img src="Images/top/menu_04.gif" name="image4" width="94" height="119" border="0" onMouseMove="move(this,'4')" onMouseout="out(this,'4')">
<img src="Images/top/menu_05.gif" name="image5" width="95" height="119" border="0" onMouseMove="move(this,'5')" onMouseout="out(this,'5')">
```

```
<img src="Images/top/menu_06.gif" name="image6" width="94" height="119" border="0" onMouseMove="move(this,'6')" onMouseout="out(this,'6')">
<img src="Images/top/menu_07.gif" width="95" name="image7" height="119" border="0" onMouseMove="move(this,'7')" onMouseout="out(this,'7')">
```

（3）编写自定义的 JavaScript 函数 move()和 out()，move()用于设置鼠标移入导航按钮时显示的图片，out()用于设置鼠标移出导航按钮时显示的图片，代码如下：

```
<script language="javascript">
//鼠标移动效果
var A_Img=new Image();
function move(image,num){
 image.src='Images/top/menu_0'+num+'_over.gif';
}
function out(image,num){
image.src='Images/top/menu_0'+num+'.gif';
}
</script>
```

秘笈心法

心法领悟 222：Image 构造器。

用 Image()构造器创建图形的语法格式如下：

```
ObjImg=new image()                          //创建了一个新的 Image 对象，并将其赋予变量 ObjImg
ObjImg.src="图片文件相对路径"                 //设置 Image 对象的 Src 属性
```

在利用 Image 对象编程之前，利用<img>标记显示属性。在 HTML 中，要显示图片可以用如下语句来实现：

```
<img src="pictureName.gif" name="ImgName">
```

在 JavaScript 中可以用下面的方式访问 Image 对象：

```
document.Img.src=" menu_01.gif";
```

第 2 篇

常用技术篇

第 4 章

文件操作

- 文件上传
- 文件下载
- 文件遍历
- 文件操作
- 文件的压缩与解压
- 目录操作

4.1　文 件 上 传

文件上传是网站中一个非常重要的功能，特别是在网站的后台管理系统中，需要通过此功能向网站中上传很多数据，包括图片、文件等。

实例 223　以二进制形式上传文件到数据库

光盘位置：光盘\MR\04\223

中级

趣味指数：★★☆

实例说明

文件的上传可以分为两类：一类是上传文件到服务器，即存储到服务器指定的文件夹下；另一类是将数据以二进制的形式上传到数据库中。具体采用哪种方式，需要根据程序的实际需要而定。

本实例中将介绍如何将文件以二进制的形式上传到数据库中，并且输出上传的数据。运行结果如图 4.1 所示。

图 4.1　将文件以二进制的形式上传到数据库

关键技术

本实例主要应用表单中的文件域提交图片文件代码如下：

```
<input name="cover" type="file" id="cover" size="30">
```

其中，参数 type 指定表单元素的类型为文件域；name 指定表单元素的名称；size 指定表单元素的大小。

通过$_POST[]方法获取表单中提交的数据，并应用 insert 语句将表单中的数据添加到数据表中。Insert 语句的语法如下：

```
Insert into table_name (column_name,column_name2, … ) values (value1, value2, … )
```

参数说明：

table_name：数据表的名称。

column：数据表中字段的名称。

valuel：表单提交的变量。

在 index_ok.php 文件中，通过 fopen()函数和 fread()函数读取表单中提交的图片数据，并且将返回的数据存储到数据库中。

在这里要特别注意网页编码格式的使用。如果网页的编码格式是 UTF-8，那么在获取上传文件的存储路径时，必须将 UTF-8 编码格式的字符串转换为 gb2312 编码格式，否则文件不能被 fopen()函数读取。

设计过程

（1）通过 Dreamweaver 开发工具创建一个 index.php 页，添加一个表单，设置表单的 method 属性值为 post，设置表单的 Action 属性值为 index_ok.php；添加表单元素，提交图书信息；添加提交按钮，其关键代码如下：

```
<form name="form" method="post" action="index_ok.php" onSubmit="return check_form(this)">
<input name="cover" type="file" id="cover" size="30">
<input type="submit" name="Submit" value="提交">
</form>
```

（2）创建 conn 文件夹，编写 conn.php 文件，连接 MySQL 数据库服务器，连接 db_database04 数据库，设置数据库变量格式为 utf8。

（3）创建 index_ok.php 文件，通过$_POST[]方法获取表单中提交的数据，并判断上传图片的格式是否正确，通过 fopen()函数和 fread()函数读取表单中提交的图片数据，编写 insert 语句将数据添加到指定的数据表中，其关键代码如下：

```
<?php
include("conn/conn.php");                                                   //连接数据库
$bookname=$_POST["bookname"];                                               //获取表单中提交的数据
$price=$_POST["price"];
$maker=$_POST["maker"];
$issuDate=$_POST["issuDate"];
$publisher=$_POST["publisher"];
$synopsis=$_POST["synopsis"];
$cover=$_POST["cover"];                                                     //获取表单中提交的图片
if($_POST["Submit"]==true){
    $cover_type=strstr($cover,".");                                         //获取从“.”到最后的字符
    if($cover_type != ".jpg" && $cover_type != ".gif" && $cover_type != ".JPG" && $cover_type != ".GIF" && $cover_type != ".bmp" &&
$cover_type != ".BMP"){                                                     //判断图片的格式
        echo "<script>alert('封面图片格式不对，请进行处理后在上传！'); window.location.href='index.php';</script>";
    }else{
$cover=iconv("utf-8","gb2312",$cover);                                      //设置字符串的编码格式
        $fp=fopen($cover,"rb");                                             //以二进制形式打开图片
        $image=addslashes(fread($fp,filesize($cover)));                     //读取二进制的数据
        $sql="insert into tb_book(bookname,price,maker,issuDate,publisher,synopsis,cover) values('$bookname',
'$price','$maker','$issuDate','$publisher','$synopsis','$image')";          //将数据添加到指定的数据表中
        $result=mysql_query($sql,$conn);
        //省略了部分代码
    }
}
?>
```

秘笈心法

心法领悟 223：读取数据库中存储的二进制图片数据。

读取数据库中存储的二进制图片数据需要定义一个单独的文件，根据数据的 ID 值执行查询，并应用 mysql_reult()函数返回查询结果，最后输出查询结果。本实例中定义的是 images.php 文件，通过该文件完成对数据库中存储的二进制图片的读取操作。images.php 的代码如下：

```
<?php
include("conn/conn.php");                                                   //连接数据库
    $query="select * from tb_book where id=".$_GET[recid];                  //定义查询指定数据的语句
    $result=mysql_query($query);                                            //执行查询
    if(!$result) die("error: mysql query");                                 //返回失败的结果
```

```
    $num=mysql_num_rows($result);                          //获取查询结果的数量
    if($num<1) die("error: no this recorder");             //如果查询结果小于则说明没有数据
    $data = mysql_result($result,0,"cover");               //获取查询结果
    mysql_close($conn);                                    //关闭数据库
    echo $data;                                            //输出返回的数据
?>
```

实例 224　**上传文件到服务器**　高级

光盘位置：光盘\MR\04\224　趣味指数：★★★★☆

实例说明

在开发网站的过程中，不但可以将文件上传到数据库中，而且可以将文件上传到服务器，这种方式可以为网站中的数据库节省很多空间，同时可以控制上传文件的大小、格式，并且读取服务器中的文件要比从数据库中读取方便很多。运行本实例，如图 4.2 所示，单击图中的“浏览”按钮，选择要上传的文件，然后单击“提交”按钮，如果上传成功，则在页面中提示“上传成功！”，否则给出错误提示信息。



图 4.2　上传文件

关键技术

PHP 中，应用 move_uploaded_file()函数实现将文件上传到指定文件夹。但是，在执行文件上传之前，为了防止潜在的攻击对原本不能通过脚本交互的文件进行非法管理，可以先应用 is_uploaded_file()函数判断指定的文件是否是通过 HTTP POST 上传，如果是则返回 TRUE。is_uploaded_file()函数的语法如下：

```
bool is_uploaded_file ( string filename )
```

参数 filename 必须指定类似于$_FILES['filename']['tmp_name']的变量，不可以使用从客户端上传的文件名$_FILES['filename']['name']。

通过 is_uploaded_file()函数对上传文件进行判断，可以确保恶意的用户无法欺骗脚本去访问本不能访问的文件，例如/etc/passwd。

move_uploaded_file()函数将文件上传到服务器中指定的位置。如果成功返回 TRUE，否则返回 FALSE，其语法如下：

```
bool move_uploaded_file ( string filename, string destination )
```

参数 filename 指定上传文件的临时文件名，即$_FILES[tmp_name]；参数 destination 指定文件上传后保存的新路径和名称。

说明：如果参数 filename 不是合法的上传文件，不会出现任何操作，move_uploaded_file()将返回 FALSE。如果参数 filename 是合法的上传文件，但出于某些原因无法移动，也不会出现任何操作，move_uploaded_file()将返回 FALSE，此外还会发出一条警告。

设计过程

（1）创建 index.php 文件，添加表单，设置 enctype 属性值为"multipart/form-data"，添加文件域、提交按钮，使用 POST 方法，将表单中数据提交到本页。

（2）通过$_FILES 获取上传文件的相关信息，通过 move_upload_file()函数完成图片的上传，代码如下：

```
<?php
if(!empty($_FILES[up_picture][name])){                              //判断上传内容是否为空
 if($_FILES['up_picture']['error']>0){                               //判断文件是否可以上传到服务器
        echo "上传错误:";
        switch($_FILES['up_picture']['error']){
                case 1:
                        echo "上传文件大小超出配置文件规定值";
                break;
                case 2:
                        echo "上传文件大小超出表单中约定值";
                break;
                case 3:
                        echo "上传文件不全";
                break;
                case 4:
                        echo "没有上传文件";
                break;
        }
}else{
        if(!is_dir("./upfile/")){                                   //判断指定目录是否存在
                mkdir("./upfile/");                                 //创建目录
        }
        $path='./upfile/'.time().strstr($_FILES['up_picture']['name'],'.');    //定义文件名称和存储位置
        if(is_uploaded_file($_FILES['up_picture']['tmp_name'])){               //是否是 HTTP POST 上传
                if(!move_uploaded_file($_FILES['up_picture']['tmp_name'],$path)){   //执行上传
                        echo "上传失败";
                }else{
                        echo "文件".time().$_FILES['up_picture']['name']."上传成功，大小为：".$_FILES['up_picture']['size'];
                }
        }else{
                echo "上传文件".$_FILES['up_pictute']['name']."不合法！ ";
        }
}
}
?>
```

秘笈心法

心法领悟 224：通过表单中的隐藏域控制上传文件的大小。

如果应用$_FILES['picture']['error']对上传文件的信息进行判断，当其返回值为 2 时，说明上传文件的大小超出了表单中约定的范围，而表单中的这个约定值是通过隐藏域 MAX_FILE_SIZE 的值来控制的。在设置这个隐藏域时，必须将其放置在文件域之前，否则不起作用。

实例 225	限制上传文件的大小 光盘位置：光盘\MR\04\225	高级 趣味指数：★★★★☆

实例说明

在网站开发的过程中，为了确保充分地利用服务器的空间，在开发上传功能时最好能够对上传文件的大小进行控制。本实例开发的上传功能就可以对上传文件的大小进行控制。如果上传文件的大小超过指定范围，那

么将给出提示信息，并终止上传。运行效果如图 4.3 所示。



图 4.3　限制上传文件的大小

关键技术

控制上传文件的大小有两个关键点：第一点是 PHP 的配置文件 php.ini 对上传文件的控制，如果上传文件的大小超过其指定的范围，那么上传就会执行失败；第二点是在 PHP 配置文件允许的范围内，在程序中对上传文件大小的控制。

（1）在 PHP 中，通过 php.ini 文件对上传文件进行控制，包括是否支持上传、上传文件的临时目录、上传文件的大小、指令执行的时间、指令分配的内存空间。

在 php.ini 中，定义到 File Uploads 项，完成对上传相关选项的设置。上传相关选项的含义如下：

☑ file_uploads：如果值是 on，说明服务器支持文件上传；如果为 off，则不支持。一般默认是支持的，不需修改。

☑ upload_tmp_dir：上传文件的临时目录。在文件被成功上传之前，首先存放到服务器端的临时目录中。多数使用系统默认目录，但是也可以自行设置。

☑ upload_max_filesize：服务器允许上传文件的最大值，以 MB 为单位。系统默认为 2MB，如果网站需要上传超过 2MB 的数据，那么就要修改这个值。

上述是 php.ini 的 File_Uploads 项中与上传相关选项参数的设置说明，除了 File_Uploads 项中的内容外，在 php.ini 中还有其他几个选项会影响到文件的上传。

☑ max_execution_time：PHP 中一个指令所能执行的最大时间，单位是秒。在上传超大文件时必须修改该选项，否则即使上传文件在服务器允许的范围内，若是超过了指令所能执行的最大时间，仍然无法实现上传。

☑ memory_limit：PHP 中一个指令所分配的内存空间，单位是 MB，其大小同样会影响到超大文件的上传。

（2）在客户端控制上传文件，应用的是 form 表单中的 enctype 和 method 属性，以及隐藏域 MAX_FILE_SIZE。

☑ enctype="multipart/form-data"，指定表单编码数据方式。

☑ method="post"，指定数据的传输方式。

☑ <input type="hidden" name="MAX_FILE_SIZE" value="10000" />，通过隐藏域控制上传文件的大小（单位为字节），该值不能超过 php.ini 配置文件中 upload_max_filesize 选项设置的值。它不能完全控制上传文件的大小，只是可以避免一些不必要的麻烦。

上述两种判断的结果都可以通过全局变量$_FILES 的返回值来体现。$_FILES 是一个数组，包含所有上传文件的信息。下面介绍一下$_FILES 数组中每个元素的含义，如表 4.1 所示。

表 4.1　$_FILES 数组中的元素

元　素　名	说　　明
$_FILES[filename][name]	存储上传文件的文件名，如：text.txt、title.jpg 等
$_FILES[filename][size]	存储文件大小，单位为字节
$_FILES[filename][tmp_name]	存储文件在临时目录中使用的文件名。因为文件在上传时，首先要将其以临时文件的身份保存在临时目录中
$_FILES[filename][type]	存储上传文件的 MIME 类型，MIME 类型规定各种文件格式的类型。每种 MIME 类型都是由“/”分隔的主类型和子类型组成。例如 image/gif，主类型为“图像”，子类型为 GIF 格式的文件；text/html 代表文本的 HTML 文件
$_FILES[filename][error]	存储与文件上传相关的错误代码。此项目是 PHP 4.2.0 版本中新增的内容，其返回值有 5 种，如下所示： 0：表示没有任何错误，文件上传成功 1：表示上传文件的大小超出了 PHP 配置文件指令 upload_max_filesize 选项限制的值 2：表示上传文件的大小超出了 HTML 表单中 MAX_FILE_SIZE 选项所指定的值 3：表示文件只被上传了一部分 4：表示没有上传任何文件

设计过程

（1）创建 index.php 文件。

（2）创建 index.php 文件，添加表单，设置 enctype 属性值为"multipart/form-data"，添加文件域、提交按钮，使用 POST 方法将表单中数据提交到本页。

（3）通过$_FILES 获取上传文件的相关信息，通过 move_upload_file()函数完成图片的上传，代码如下：

```
<?php
if(!empty($_FILES[up_picture][name])){                                          //判断上传内容是否为空
 if($_FILES['up_picture']['error']>0){                                          //判断文件是否可以上传到服务器
        echo "上传错误:";
        switch($_FILES['up_picture']['error']){
                case 1:
                        echo "上传文件大小超出配置文件规定值";
                break;
                case 2:
                        echo "上传文件大小超出表单中约定值";
                break;
                case 3:
                        echo "上传文件不全";
                break;
                case 4:
                        echo "没有上传文件";
                break;
        }
 }else{
        if(!is_dir("./upfile/")){                                               //判断指定目录是否存在
                mkdir("./upfile/");                                             //创建目录
        }
        $path='./upfile/'.time().strstr($_FILES['up_picture']['name'],'.');     //定义文件名称和存储位置
        if(is_uploaded_file($_FILES['up_picture']['tmp_name'])){                //是否是 HTTP POST 上传
                if(!move_uploaded_file($_FILES['up_picture']['tmp_name'],$path)){   //执行上传
                        echo "上传失败";
                }else{
                        echo "文件".time().$_FILES['up_picture']['name']."上传成功，大小为：".$_FILES['up_picture']['size'];
                }
        }else{
                echo "上传文件".$_FILES['up_pictute']['name']."不合法！";
        }
 }
```

```
}
?>
```

秘笈心法

心法领悟 225：如何上传超大文件。

如果要上传超大文件，必须修改 php.ini 文件。修改选项包括：upload_max_filesize 的最大值、max_execution_time 一个指令所能执行的最大时间和 memory_limit 一个指令所分配的内存空间。

实例 226	限制上传文件的类型 光盘位置：光盘\MR\04\226	中级 趣味指数：★★★☆

实例说明

在开发文件上传功能时，不只要考虑上传文件的大小，有时还要考虑上传文件的类型，针对不同的需要对上传文件的类型加以限制，这里指的文件类型可以通过文件的扩展名（例如.txt、.php、.doc）来判断。例如，在进行文件上传时，涉及到文本说明性文字时，最好将上传的文件限制为以“.txt”为后缀的文本格式。

运行本实例，如图 4.4 所示，单击图中的“浏览”按钮，选择要上传的文件（文件不是以“.txt”为扩展名），单击“提交”按钮，程序将给出错误提示。



图 4.4　限制上传文件的类型

关键技术

文件的上传和对大小的限制已经在前面的实例中讲解过，这里只讲解限制上传文件类型的关键技术，首先要通过预定义全局变量$_FILES 中的$_FILES[files][name]获取上传文件的名称，然后应用 strstr()截取上传文件名称的后缀，最后判断上传文件的后缀是否符合要求。

strstr()函数的语法如下：

```
string strstr(string strings, string needle);
```

该函数用于在字符串 strings 中查找字符串 needle 的位置，并返回从 needle 字符串开始到 strings 字符串结束的所有内容，如果在 strings 中没有查找到 needle，该函数将返回 FALSE。

设计过程

（1）继续使用实例 225 的内容，在其基础上增加对上传文件类型的判断，增加的关键代码如下：

```
<?php
if(!empty($_FILES[up_picture][name])){                //判断上传内容是否为空
$type=$_FILES['up_picture']['name'];                  //获取上传文件的名称
$types=strstr($type,'.');                             //截取上传文件的后缀
if($types==".txt"){                                   //判断上传文件的后缀是否符合要求
        //省略了部分代码
}
}else{
```

```
            echo "上传文件".$_FILES['up_pictute']['name']."类型不正确！";
}
}
?>
```

（2）重新保存 index.php 文件。

秘笈心法

心法领悟 226：判断上传的文件是否符合多种类型中的一种。

判断上传的文件是否符合多种类型中的一种，最佳的方法就是将多种类型定义到一个数组中，然后通过 for 循环输出数组中的元素值，将输出的元素值与获取的文件后缀进行比较，进而判断其是否符合要求。

实例 227 上传多个文件到服务器

光盘位置：光盘\MR\04\227

高级

趣味指数：★★★★☆

实例说明

上传图片到服务器，是程序开发过程中必不可少的一个功能。它不但可以达到图片共享的目的，而且可以提高网站的访问量，丰富网站的内容。在本实例中，讲解如何通过 POST 方式实现多图片上传，运行结果如图 4.5 所示。



图 4.5 POST 方式实现多图片上传

关键技术

多文件上传的关键是如何定义上传文件元素的名称及如何判断上传文件的数量。在本实例中，以数组的形式定义上传文件的名称（上传文件的名称是“files[]”）。为了达到可以上传任意数量图片（4 个图片以内）的目的，在对上传文件进行处理的过程中应用 array_filter()函数和回调函数去除数组中的空元素。

array_filter()函数，用回调函数过滤数组中的单元，语法如下：

```
array array_filter ( array input [, callback callback] )
```

array_filter()函数依次将 input 数组中的每个值传递到 callback 函数。如果 callback 函数返回 TRUE，则 input 数组的当前值会被包含在返回的结果数组中，并且数组的键名保留不变。

说明：在回调函数中不要对数组进行修改操作，例如，增加或者删除数组中的元素，如果一旦数组改变，那么此函数的运用也就没有意义了。如果没有提供 callback()函数，array_filter()将删除 input 中所有

等值为 FALSE 的元素。

本实例中定义的回调函数是 check()，用于验证数组中的元素值是否为空，其语法如下：

```
function check($var) { //验证数组的返回值是否为空
        return ($var != "");
}
```

说明：通过 POST 方法实现多图片上传，在创建 form 表单时，必须指定 enctype="multipart/form-data"属性。如果要通过隐藏域 MAX_FILE_SIZE 的值对上传文件的大小进行控制，那么必须将隐藏域放置在上传文件的文件域之前，否则是不会起作用的。

设计过程

（1）创建 index.php 文件。添加表单，设置文件域、提交按钮，使用 POST 方法，设置 enctype="multipart/form-data"，将数据提交到 index_ok.php 页，完成多个文件的上传操作，其关键代码如下：

```
<table width="750" border="0" cellspacing="0" cellpadding="0">
    <form action="index_ok.html" method="post" enctype="multipart/form-data" name="form1">
        <tr>
          <td width="100" height="25" align="right" class="STYLE1">内容 1：</td>
          <td width="150" align="center"><input name="files[]" type="text" id="files[]" size="15"></td>
          <td align="left"><input name="picture[]" type="file" id="picture[]" size="30"></td>
          </tr>
        <tr>
          <td height="25" align="right" class="STYLE1">内容 2：</td>
          <td align="center"><input name="files[]" type="text" id="files[]" size="15"></td>
          <td align="left"><input name="picture[]" type="file" id="picture[]" size="30"></td>
          </tr>
        <tr>
          <td colspan="3" align="center">
              <input type="image" name="imageField" src="images/bg_09.jpg">    
              <input type="image" name="imageField2" src="images/bg_11.jpg"></td>
          </tr>
      </form>
    </table>
```

（2）在 index.php 文件中，连接数据库，读取数据库中存储的数据，实现上传文件的分页输出。代码请参考光盘中的相关内容。

（3）创建 index_ok.php 文件，获取表单中提交的数据，将多个文件存储到服务器中，将文件的名称和存储路径存储到数据库中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include "conn/conn.php";                                                 //包含数据库链接文件
if ($_POST [files] != "") {
        if (! is_dir ( "./upfile" )) {
                mkdir ( "./upfile" );                                    //创建上传文件存储文件夹
        }
        $data = date ( "Y-m-d H:m:s" );                                  //定义时间
        function check($var) {                                           //验证数组的返回值是否为空
                return ($var != "");                                     //如果不为空则返回数组元素
        }
        $files = array_filter ( $_POST ["files"], "check" );             //去除数组中空值
        $array = array_filter ( $_FILES ["picture"] ["name"], "check" ); //去除数组中空值
        foreach ( $array as $key => $value ) {                           //循环读取数组中数据
                $path = 'upfile/' . time () . $key . strtolower ( strstr ( $value, "." ) );   //定义上传文件存储位置
                move_uploaded_file ( $_FILES ["picture"] ["tmp_name"] [$key], $path );  //执行上传操作
                $query = "insert into tb_up_file (file_test,data,file_name) values ('$path','$data','$files[$key]')";
                $result = mysql_query ( $query );
        }
        echo "<script>alert('图片上传成功'); window.location.href='index.html';</script>";
}
?>
```

秘笈心法

心法领悟 227：通过伪静态技术隐藏 PHP 文件后缀。

首先，修改 Apache 服务器的配置文件 httpd.conf。打开 httpd.conf 文件，定位到如下位置：

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

将该项前面的“#”去掉后启动该项。

然后，查找 httpd.conf 文件，找到其中的 AllowOverride 项，将它的值都修改为 All。保存并重新启动 Apache 服务器，使修改生效。

最后，在实例根目录下创建.htaccess 文件，实现对 PHP 文件后缀的隐藏操作。.htaccess 文件的代码如下：

```
RewriteEngine On            #启动项
RewriteRule ^index.html$ index.php
RewriteRule ^index  ok.html$ index  ok.php
RewriteRule ^index-([0-9]+)-([0-9]+)-([0-9]+)\.html$ index.php?vv=$1&ljjl=$2&page=$3 [L]
```

通过正则表达式对文件的后缀和传递的参数进行匹配，完成对 PHP 文件后缀的隐藏操作。

4.2 文 件 下 载

实例 228　通过链接方式下载

光盘位置：光盘\MR\04\228

初级

趣味指数：★★☆

实例说明

在一个完整的网站中，具有文件下载功能能够给网站带来更多的访问者，增加网站的访问量。最常用的文件下载方式就是通过链接下载。在本实例中，应用链接方式进行下载，当单击“下载”超链接时，即可弹出文件下载的对话框，完成下载操作，运行结果如图 4.6 所示。



图 4.6　链接方式的文件下载

关键技术

本实例主要通过超链接来实现文件的下载。通过 select 语句从数据库中读取文件存储的路径，然后将该文件作一个超链接，当单击“下载”超链接时，即可下载该文件。

设计过程

（1）在实例 227 的基础上，为输出的图片名称创建超链接，完成图片的下载操作，其关键代码如下：

```
<a href="<?php echo $myrow[file_test];?>"><?php echo $myrow[file_name];?></a>
```

（2）保存对文件的修改，运行程序，运行结果参看实例说明。

秘笈心法

心法领悟 228：不同类型的文件，下载操作不同。

如果是压缩的文件可以直接下载；如果不是，则需要单击鼠标右键选择“另存为”命令。

实例 229　通过 header 函数进行下载

光盘位置：光盘\MR\04\229

中级

趣味指数：★★★★☆

实例说明

除了可以通过链接方式下载文件之外，还可以通过 header()函数完成下载操作。例如，本实例中就是应用 header()函数实现文件的下载，如果下载的文件不存在，则会给出提示信息，运行结果如图 4.7 所示。

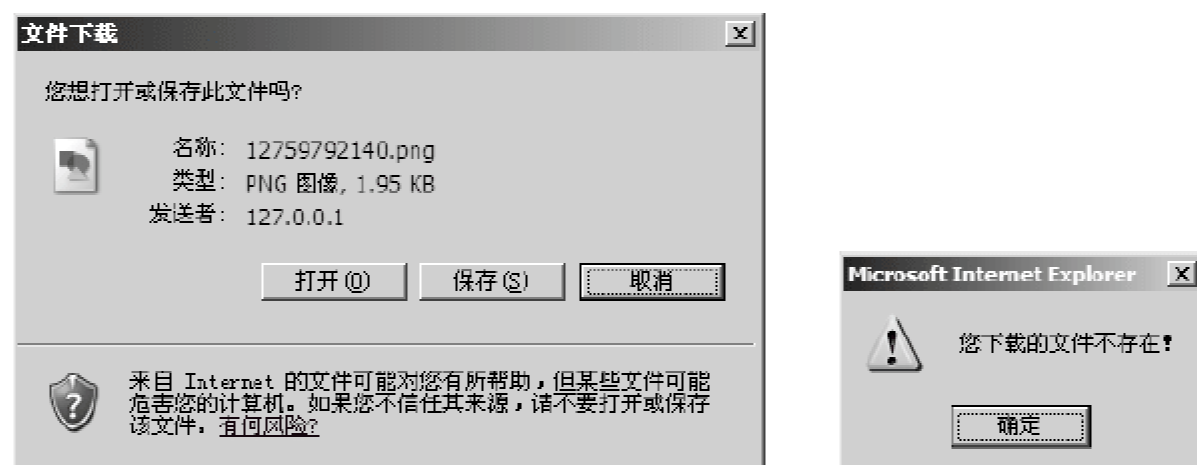


图 4.7　header()函数实现文件下载

关键技术

通过 HTTP 方式下载文件，主要应用 header()函数。

header()函数属于 HTTP 函数，其作用是以 HTTP 协议将 HTML 文档的标头送到浏览器，并告诉浏览器具体怎么处理这个页面。header()函数的语法如下：

```
void header ( string string [, bool replace [, int http_response_code]] )
```

参数说明：

string：发送的标头。

replace：如果一次发送多个标头，对于相似的标头是替换还是添加。如果是 false，则强制发送多个同类型的标头。默认是 true，即替换。

http_response_code：强制 HTTP 响应为指定值。

通过 HTTP 下载的代码如下：

```
header("Content-type: application/x-gzip");
header("Content-Disposition: attachment; filename=文件名");
header("Content-Description: PHP3 Generated Data"); >
```

HTTP 标头有很多，这里介绍的是下载的 HTTP 标头，其代码如下：

```
header('Content-Disposition: attachment; filename="filename"');
```

在应用过程中，唯一需要改动的就是 filename，即将 filename 替换为要下载的文件。

设计过程

（1）在实例 227 的基础上，为输出的图片创建超链接，链接到 download.php 文件，通过 download.php 完成文件的下载操作。在 index.php 文件中所做的修改如下：

```
<td width="245" height="100" align="center" valign="middle" bgcolor="#F0F0F0"><a href="download-<?php echo $myrow[file_test];?>.html" title="点击即可下载！"><?php echo "<img src=\"$myrow[file_test]\" width=\"250\" height=\"100\" border=\"0\">"?></a></td>
```

（2）创建 download.php 文件，应用 header()函数，实现文件的下载，其代码如下：

```
<?php
header("Content-type: text/html; charset=UTF-8");                    //设置文件编码格式
$path = $_GET['path'];                                               //获取文件路径
if(!empty($path) and !is_null($path)){                               //判断变量是否为空，是否为 NULL
    $filename=basename($path);                                       //获取文件名
    $file=@fopen($path,"r");
    if($file){
        header("Content-type:application/octet-stream");             //输出 MIME 类型
        header("Accept-ranges:bytes");                               //接受的范围单位
        header("Accept-length:".filesize($path));                    //文件长度
        header("Content-Disposition:attachment;filename=".$filename);//缺省时文件保存对话框中的文件名称
        echo fread($file,filesize($path));                           //读取文件
        fclose($file);                                               //关闭文件
        exit;                                                        //退出
    }else{
        echo "<script>alert('您下载的文件不存在！'); history.back();</script>";
    }
}
?>
```

秘笈心法

心法领悟 229：header()函数的应用。

通过 header()函数，不但可以实现文件的下载，还可以实现以下 3 个功能：

（1）重定向，这是最常用的功能。

```
header("Location: http://www.mrbccd.com");
```

（2）强制客户端每次访问页面时获取最新资料，而不是使用存在于客户端的缓存。

```
//设置页面的过期时间(用格林威治时间表示)。
header("Expires:    Mon,    08    Jul    2008    08:08:08    GMT");
//设置页面的最后更新日期(用格林威治时间表示)，使浏览器获取最新资料
header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "GMT");
header("Cache-Control: no-cache,    must-revalidate");          //控制页面不使用缓存
header("Pragma: no-cache");                                     //参数（与以前的服务器兼容），即兼容 HTTP1.0 协议
header("Content-type:    application/file");                    //输出 MIME 类型
header("Content-Length:    227685");                            //文件长度
header("Accept-Ranges:    bytes");                              //接受的范围单位
//缺省时文件保存对话框中的文件名称
header("Content－Disposition: attachment; filename=$filename");  //实现下载
```

（3）输出状态值到浏览器，控制访问权限。

```
header('HTTP/1.1 401 Unauthorized');
header('status: 401 Unauthorized');
```

4.3 文 件 遍 历

实例 230 | 读取整个文件的内容 | 初级
光盘位置：光盘\MR\04\230 | 趣味指数：★★☆

实例说明

在开发网站的过程中，很多的服务条款、协议等都是以文本文件的形式存储的。如果要读取这些文件中的内容，就需要应用到文件系统函数。在本实例中，将介绍如何通过文件系统函数读取整个文件的内容，其运行效果如图 4.8 所示。

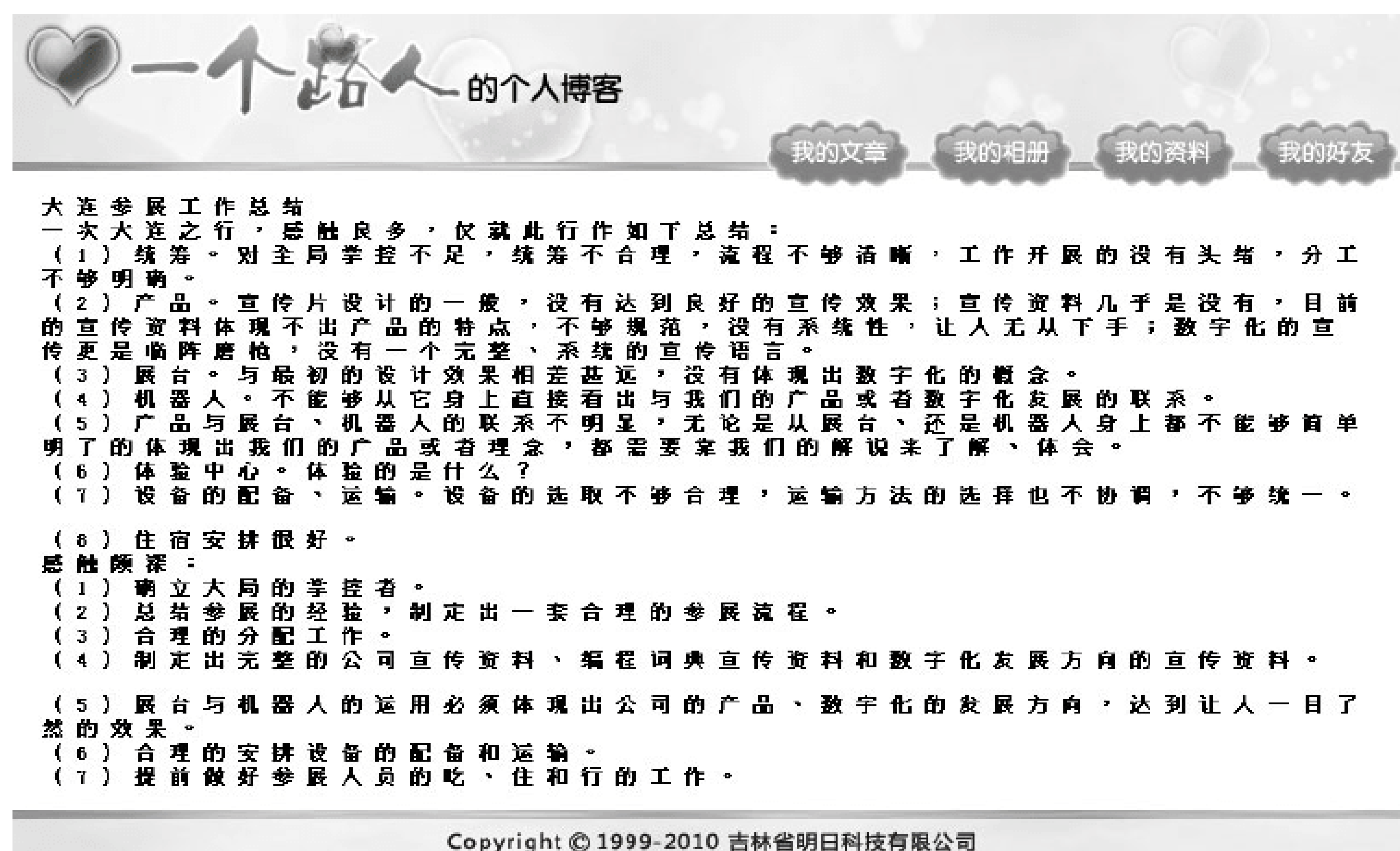


图 4.8　读取文本文件的内容

关键技术

在 PHP 中，读取整个文件的内容，有 3 个函数可以选择：

☑　readfile()函数

readfile()函数读取一个文件并将其写入输出缓冲，成功，返回读取的字节数；失败，返回 FALSE，语法如下：

```
int readfile ( string filename [, bool use_include_path [, resource context]] )
```

参数 filename 指定读取的文件名称；参数 use_include_path 控制是否支持在 include_path 中搜索文件，如果支持，则将该值设置为 TRUE；参数 context 是 PHP 5.0 的新增内容。

说明： *应用 readfile()函数时，不需要打开/关闭文件和输出语句，直接应用即可。*

☑　file()函数

file()函数将整个文件的内容读入一个数组中。成功，返回数组，数组中的每个元素都是文件中对应的一行，包括换行符在内；失败，返回 FALSE。语法如下：

```
array file ( string filename [, int use_include_path [, resource context]] )
```

其参数与 readfile()函数相同，唯一区别是该函数的返回值是数组。

☑　file_get_contents()函数

file_get_contents()函数将文件内容读入一个字符串。如果有 offset 和 maxlen 参数，将在参数 offset 所指定的位置开始读取长度为 maxlen 的内容。如果失败，返回 FALSE。语法如下：

```
string file_get_contents ( string filename [, bool use_include_path [, resource context [, int offset [, int maxlen]]]] )
```

参数 filename 指定读取的文件名称；参数 use_include_path 控制是否支持在 include_path 中搜索文件，如果支持，则将该值设置为 TRUE。

设计过程

（1）创建 index.php 文件。

（2）在 index.php 文件中分别应用 file()和 file_get_contents()函数读取文本文件中的内容，其关键代码如下：

```
<!--  使用 file()函数读取 count.txt 文件的内容  -->
<?php
$arr = file('test1.txt');
foreach ($arr as $value) {
      echo iconv("gb2312", "utf-8", $value) . "<br>";
}
```

```
?>
<!--   使用 file_get_contents()函数读取 count.txt 文件的内容   -->
<?php
$str = file_get_contents('test2.txt');
echo iconv("gb2312", "utf-8", $str);
?>
```

秘笈心法

心法领悟 230：读取整个文件的特性。

在通过 readfile()、file()和 file_get_contents()函数读取整个文件中的内容时，不需要通过 fopen()函数打开文件，也不需要使用 fclose()函数关闭文件。

但是，在读取一个字符、一行字符和任意长度的字符串时，必须应用 fopen()函数打开文件后才能进行读取，在读取完成后还要应用 fclose()函数关闭文件。

实例 231	文本文件的分页读取 光盘位置：光盘\MR\04\231	高级 趣味指数：★★★★☆

实例说明

在遍历文件中的内容时，由于文件内容很多，最理想的方法就是分页读取文本文件中的内容。本实例将介绍如何分页读取文本文件中的数据，其运行结果如图 4.9 所示。



图 4.9 文本文件的分页读取

关键技术

完成超长文本的分页输出需要 3 方面的技术：第 1 方面，自定义函数。通过自定义函数读取文本文件，可以避免中文字符串出现乱码；第 2 方面，字符串函数。需要通过 strlen()函数计算字符串的长度，通过 substr()函数对字符串进行截取；第 3 方面，文件系统函数。通过 file_get_contents()函数读取文本文件中的数据。

自定义函数 msubstr()的语法如下：

```
//定义一个用于截取一段字符串的函数 msubstr()
function msubstr($str,$start,$len){                //$str 指的是字符串，$start 指的是字符串的起始位置，$len 指的是长度
$strlen=$start+$len;                               //用$strlen 存储字符串的总长度（从字符串的起始位置到字符串的总长度）
for($i=0;$i<$strlen;$i++){                         //通过 for 循环语句，循环读取字符串
    if(ord(substr($str,$i,1))>0xa0){               //如果字符串中首个字节的 ASCII 序数值大于 0xa0，则表示为汉字
        $tmpstr.=substr($str,$i,2);                //每次取出两位字符赋给变量$tmpstr，即等于一个汉字
        $i++;                                      //变量自加 1
```

```
        }else{                              //如果不是汉字，则每次取出一位字符赋给变量$tmpstr
            $tmpstr.=substr($str,$i,1);
}
}
return $tmpstr;                             //输出字符串
}
```

设计过程

（1）创建 function.php 文件，编写自定义函数 msubstr()，完成对文本文件的截取操作。

（2）创建 index.php 文件，首先通过文件系统函数 file_get_contents()读取整个文件的内容，然后调用自定义函数和字符串函数完成对文件的截取操作，实现截取后内容的分页输出，其关键代码如下：

```
<div id="synopsis">
    <!--创建 div 标签，用于获取 js 文件中返回的分页结果-->
    <table width="545" border="0" cellspacing="0" cellpadding="0">
        <tr>
            <td><?php
                    include("function.php");
//读取超长文本中的数据，实现超长文本中数据的分页显示
  if($_GET['page']){
 $counter=file_get_contents("data.txt");
     $length=strlen($counter);
     $page_count=ceil($length/850);
     $c=msubstr($counter,0,($_GET['page']-1)*850);
     $c1=msubstr($counter,0,$_GET['page']*850);
      echo substr($c1,strlen($c),strlen($c1)-strlen($c));
  }
?>
            </td>
        </tr>
        <tr>
            <td><table width="535" border="0" align="center" cellpadding="0" cellspacing="0">
                    <tr>
                        <!--设置超长文本分页显示的超级连接-->
                        <td width="37%" align="center" valign="middle" bgcolor="#FFFFFF">内容 <?php echo $page_count;?> 页
  当前 第 <?php echo $_GET['page'];?> 页</td>
                        <td width="63%" height="28" align="center" valign="middle" bgcolor="#FFFFFF"><?php
              if($_GET['page']!=1){
?>
                            <!--调用 no_refurbish_pagination 函数，实现无刷新的分页输出-->
                            <a href="#" onClick='return no_refurbish_pagination("index_ok.php?page=1")'>首页</a>  <a href="#"
onClick='return no_refurbish_pagination("index_ok.php?page=<?php echo $_GET['page']-1;?>")'>上一页</a>
                            <?php   }
                   if($_GET['page']<$page_count){
?>
                            <a href="#" onClick="return no_refurbish_pagination('index_ok.php?page=<?php echo $_GET['page']+1;?>')">下一
页</a> <a href="#" onClick='return no_refurbish_pagination("index_ok.php?page=<?php echo $page_count;?>")'>尾页</a>
                            <?php

         }
              ?>
                        </td>
                    </tr>
                </table></td>
        </tr>
    </table>
</div>
```

（3）创建 index_ok.php 文件，实现数据的无刷新分页。index_ok.php 中的内容与 index.php 中 div 标签 synopsis 的内容相同。

（4）创建 js 脚本文件 discuss_js.js，通过 Ajax 实现无刷新的操作。

秘笈心法

心法领悟 231：Ajax 无刷新分页。

在本实例中不但实现了文本文件内容的分页输出，而且是无刷新分页输出。具体操作通过 js 脚本文件 discuss_js.js 和 index_ok.php 完成。

4.4 文 件 操 作

实例 232　文件操作汇总　中级

光盘位置：光盘\MR\04\232　趣味指数：★★★★☆

实例说明

为了便于对网站进行管理、维护和更新，应该设计一个能够对文件进行操作的模块，实现对文件的创建、复制、移动和删除等操作，这样能够给网站的管理工作提供很大的方便，不再因为要修改某个文件而登录到 FTP 中、通过下载或上传实现文件的更新，从而节省很多时间。运行本实例，实现对指定文件的创建、复制、移动和删除操作。只要在文本框中输入要复制文件的路径和名称，在对应的文本框中输入指定文件要复制到的具体文件夹的路径和名称（包括指定文件的名称），然后单击“提交”按钮即可，运行结果如图 4.10 所示。



图 4.10　文件操作汇总

关键技术

本实例主要应用 fopen()、copy()、rename()和 unlink()4 个函数完成文件的创建、复制、移动和删除操作。

fopen()函数用于打开文件或者 URL，其语法如下：

```
int fopen(string filename, string mode);
```

创建文件主要应用 fopen()函数的 mode 参数，mode 参数的可选值如下：

☑ 'w' 写入方式打开，将文件指针指向文件头并将文件大小截为零。如果文件不存在，则进行创建。

☑ 'w+' 读写方式打开，将文件指针指向文件头并将文件大小截为零。如果文件不存在，则进行创建。

☑ 'a' 写入方式打开，将文件指针指向文件末尾。如果文件不存在，则进行创建。

☑ 'a+' 读写方式打开，将文件指针指向文件末尾。如果文件不存在，则进行创建。

说明：上述 4 种方式都可以完成文件的创建操作，前提是指定路径下不存在被创建的文件。

copy()函数主要用于复制文件，其语法如下：

```
int copy(string source, string dest);
```

本函数将文件从 source 复制到 dest，如果成功则返回 TRUE，否则返回 FALSE。

rename()函数主要用于文件重命名，其语法如下：

```
bool rename(string oldname,newname);
```

本函数将 oldname 重命名为 newname，成功则返回 TRUE，否则返回 FALSE。

说明： ename()函数不但可以给文件重命名，当将指定的文件移动到另外一个路径下的文件夹中时，不改变该文件的名称，即可实现对指定文件的移动操作。

unlink()函数主要用于删除文件，其语法如下：

```
bool unlink(string filename);
```

本函数用于删除文件。如果删除成功则返回 true，否则返回 false。

设计过程

（1）创建 index.php 文件。首先，设计文件汇总的页面。然后应用 switch()语句设计一个简单的框架，完成复制、移动、创建和删除操作之间的切换。最后，编写 PHP 脚本，根据表单中提交的数据，执行不同的文件操作，其关键代码如下：

```
<?php
switch ($_GET ['operate']) {
        case "found" :
                include ("found.php");
                break;
        case "copy" :
                include ("copy.php");
                break;
        case "move" :
                include ("move.php");
                break;
        case "delete" :
                include ("delete.php");
                break;
        default :
                include ("copy.php");
}
?>
<?php
if ($_POST ['Submit'] == "复制") {
$copy = iconv ( "utf-8", "gb2312", $_POST ['copys'] );
$copys2 = iconv ( "utf-8", "gb2312", $_POST ['copys2'] );
if (copy ( $copy, $copys2 )) {
        echo "<script>alert('复制成功!!');</script>";
} else {
        echo "<script>alert('复制失败!!');</script>";
}
}
//省略了部分代码
?>
```

（2）分别创建 copy.php、delete.php、found.php 和 move.php 文件，完成不同操作的执行页面。copy.php 文件的代码如下：

```
<table width="466" height="112" border="0" cellpadding="0" cellspacing="0">
<form name="form1" method="post" action="">
<tr>
        <td width="111" rowspan="2" align="center">复制文件</td>
        <td width="67" align="right" valign="bottom">copy：</td>
        <td width="229" height="39" valign="bottom"><input name="copys"
                type="text" id="copys" size="22"></td>
        <td width="126" rowspan="2" align="left" valign="middle"><input
                type="submit" name="Submit" value="复制"></td>
</tr>
```

```
<tr>
        <td align="right" valign="top">affix：</td>
        <td height="41" valign="top"><input name="copys2" type="text"
                id="copys2" size="22"></td>
</tr>
<tr>
        <td colspan="4" align="center">例如：F:\xampp\htdocs\MR\05\012\test.txt</td>
</tr>
</form>
</table>
```

秘笈心法

心法领悟 232：编码格式转换的必要性。

在应用文件操作函数执行文件的创建、移动、复制和删除操作时，由于实例的页面编码格式是 utf-8，所以，从表单中获取的元素值，不能够直接应用到文件操作函数中，必须对其进行编码转换，由 utf-8 编码转换为 gb2312 编码，否则文件操作函数不能够正确地执行。

实例 233	目录、文件定位器 光盘位置：光盘\MR\04\233	中级 趣味指数：★★★★☆

实例说明

对目录、文件的操作不应该是盲目的，应该首先确定其是否存在，如果存在才能够对它执行各种操作，否则是没有任何意义的。相反，有些操作必须是在指定的目录、文件不存在的情况下进行的。此时必须有一个方法能够对目录、文件进行定位，确定它是否存在，这就是本实例将要讲解的内容。运行本实例，在文本框中输入目录、文件的完整路径，单击“提交”按钮，运行结果如图 4.11 所示。



图 4.11　目录、文件定位器

说明：本实例实现的功能虽然非常简单，但是这个方法在很多程序中都会被用到，只是表现的方式不同。

关键技术

PHP 中判断指定的目录、文件是否存在，应用的是 file_exists()函数，函数的语法如下：

本函数用来检测目录、文件是否存在，如果存在返回 TRUE，否则返回 FALSE。

```
int file_exists(string filename);
```

说明：要在 Windows 下检查网络中的共享文件应该使用//computername/share/filename 或者\\computername\share\filename。

设计过程

（1）创建 index.php 文件，设计页面布局。

（2）添加表单，设置文本框，提交目录、文件的完整路径，并设置提交按钮。

（3）编写 PHP 脚本，获取表单中提交的目录、文件路径。首先对获取的元素值进行编码格式的转换，然后应用 file_exists()函数判断指定的目录、文件是否存在，其关键代码如下：

```
<?php
if ($_POST ['file_name'] != "") {
	$file_name = iconv ( "utf-8", "gb2312", $_POST ['file_name'] );
	if (file_exists ( $file_name )) {
		echo "<script>alert('目录、文件已定位！');</script>";
	} else {
		echo "<script>alert('目录、文件不存在！');</script>";
	}
} else {
	echo "<script>alert('请输入正确的目录、文件路径！');</script>";
}
?>
```

秘笈心法

心法领悟 233：utf-8 编码格式对中文字符串的影响。

在本实例中，由于使用的是 utf-8 编码格式，因此当提交的目录、文件路径中存在中文字符串时，file_exists()函数就不能够正确地判断目录、文件是否存在。只有将 utf-8 编码格式的元素值转换为 gb2312 编码格式之后，file_exists()函数才能对中文字符串的目录、文件路径作出正确的判断。

实例 234　改头换面　初级

光盘位置：光盘\MR\04\234　趣味指数：★★✩

实例说明

在对网站进行管理和维护的过程中，经常会修改文件的名称。本实例将介绍一种修改文件名称的方法。运行本实例，在“原文件名称”文本框中输入原始文件的存储路径和文件名，在新文件名称文本框中输入新文件的路径和文件名，单击“提交”按钮，运行结果如图 4.12 所示。



图 4.12　修改文件名

关键技术

文件重命名应用的是 rename()函数，不过首先要应用 file_exists()函数判断该文件是否存在，如果存在则应用 rename()函数对该文件进行重命名，否则给出提示信息。

设计过程

（1）创建 index.php 文件，设计页面布局。

（2）添加表单，设置两个文本框，提交文件的原始名称和新名称，并设置提交按钮。

（3）编写 PHP 脚本，获取表单中提交的元素值。由于网页使用的是 utf-8 编码，所以为了避免在获取中文字符串时出现乱码，应先用 iconv()函数对获取的元素值进行编码格式的转换，然后应用 file_exists()函数判断指定的文件是否存在，如果存在则应用 rename()函数对其进行重新命名，其关键代码如下：

```
<?php
if ($_POST['Submit'] == "提交") {
	$old_name=iconv("utf-8","gb2312",$_POST['old_name']);		//获取表单元素，并进行编码格式转换
	$new_name=iconv("utf-8","gb2312",$_POST['new_name']);		//获取表单元素，并进行编码格式转换
	if (file_exists ( $old_name )) {					//判断文件是否存在
		if (rename ( $old_name, $new_name )) {			//把原文件重新命名
			echo "修改成功!";
		}
	} else {
		print $_POST['old_name'] . "文件不存在!<br>";
	}
}
?>
```

秘笈心法

心法领悟 234：iconv()函数在转码过程中的重要性。

如果程序设计时应用的是 utf 8 编码，那么就有必要了解 iconv()函数。因为操作系统默认使用的是简体中文（gb2312 编码或者 gbk 编码），所以在将 utf 8 编码格式的数据应用到默认为 gb2312 编码的程序中时，难免会执行失败，特别是涉及到中文字符串时。例如，在本实例中，如果直接将获取的表单元素值应用到 file_exists()函数或者 rename()函数中，就会出现乱码。

实例 235	文件属性分析 光盘位置：光盘\MR\04\235	中级 趣味指数：★★★★☆

实例说明

如果说判断目录、文件是否存在是对目录、文件进行操作的前提条件，那么获取文件属性则是对文件进行操作的必要条件。因为在执行一些特殊的操作之前，必须对文件的类型、大小或者修改时间进行判断。在本实例中，将介绍如何获取文件的类型、大小和修改时间。运行本实例，在文本框中输入正确的目录或者文件的路径，单击“提交”按钮，运行结果如图 4.13 所示。



图 4.13　文件属性分析

关键技术

PHP 中，获取文件类型使用 filetype()函数，获取文件大小使用 filesize()函数，而获取修改时间使用 filemtime()函数。

filetype()函数，用于获取文件类型，其语法如下：

```
string filetype ( string filename )
```

该函数返回文件的类型，类型值包括 fifo、char、dir、block、link、file 和 unknown。

如果出错则返回 FALSE。

filesize()函数，用于获取文件大小，其语法如下：

```
int filesize ( string filename )
```

该函数返回文件大小的字节数，如果出错返回 FALSE（在错误报告级别为 E_WARNING 的情况下）。

说明： 因为 PHP 的整数类型是有符号的，并且大多数平台使用 32 位整数，filesize() 函数在碰到大于 2GB 的文件时可能会返回非预期的结果。对于 2GB~4GB 之间的文件通常可以使用 sprintf("%u", filesize($file)) 来克服此问题。

filemtime()函数，用于获取文件修改时间，其语法如下：

```
int filemtime ( string filename )
```

该函数返回文件上次被修改的时间，出错时返回 FALSE。时间以 UNIX 时间戳的方式返回，可用于 date()。

设计过程

（1）创建 index.php 文件，设计页面布局。

（2）添加表单，设置目录、文件提交的文本框，提交目录或者文件的路径，并设置提交按钮。

（3）编写 PHP 脚本，获取表单中提交的元素值。首先，判断提交的值是否为空，如果不为空，则应用 iconv()函数对字符串的编码格式进行转换。然后，应用 file_exists()函数判断指定的目录或者文件是否存在，如果存在则获取目录或者文件的类型、大小以及修改时间，其关键代码如下：

```
<?php
	if ($_POST ['file_name'] != "") {                                          //判断表单提交的值是否为空
		$file_name = iconv ( "utf-8", "gb2312", $_POST ['file_name'] );        //完成编码格式的转换
		if (file_exists ( $file_name )) {                                      //判断目录或者文件是否存在
			$file_type = filetype ( $file_name );                              //获取文件类型
			echo "<br>文件类型：" . $file_type . "<br>";
			if ($file_type != "dir") {                                         //判断如果不是目录
				$file_size = filesize ( $file_name );                          //获取文件的大小
				echo "文件大小：" . $file_size . " 字节" . "<br>";
			}
			$file_mtime = filemtime ( $file_name );                            //获取目录或者文件的修改时间
			echo "修改时间：" . date ( "Y-m-d H:i:s", $file_mtime );           //对返回的时间戳进行格式化输出
		} else {
			echo "<script>alert('目录、文件不存在！');</script>";
		}
	} else {
		echo "<script>alert('请输入正确的目录、文件路径！');</script>";
	}
?>
```

秘笈心法

心法领悟 235：获取文件的其他信息。

在 PHP 内置的文件系统操作函数中，不仅可以获取文件的类型、大小和修改时间，还可以获取文件的上次访问时间（fileatime()）、inode 修改时间（filectime()）、文件的组（filegroup()）、文件的所有者（fileowner()）、和文件的权限（fileperms()）等。

实例 236 文件类型检测

光盘位置：光盘\MR\04\236

高级

趣味指数：★★★★☆

实例说明

在开发程序的过程中，很多时候都需要获取到文件的后缀名，并根据后缀名作出一些判断。那么如何才能获取文件的后缀名呢？在本实例中将对这个问题进行解答。运行本实例，将检测表单中提交文件的后缀名，并根据后缀名称，作出相应的判断，其运行结果如图 4.14 所示。



图 4.14　上传文件类型检测

关键技术

在本实例中应用 move_uploaded_file()函数实现文件上传的操作，并根据文件的后缀名进行判断，将不同类型的文件存储在不同的服务器文件夹下。

获取上传文件后缀名时，首先，应用$_FILES 全局变量获取上传文件的名称。然后，应用 strstr()函数对上传文件的名称进行截取，截取字符串中“.”后的所有字符串。最后，应用 strtolower()函数将字符串转换成小写。

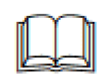
strstr()函数，用于获取一个指定字符串在另一个字符串中首次出现的位置到后者末尾的子字符串，语法如下：

```
string strstr ( string haystack, string needle)
```

参数说明：

Haystack：必要参数，指定从哪个字符串中进行搜索。

Needle：必要参数，指定搜索的对象。如果该参数是一个数值，那么将搜索与这个数值的 ASCII 值相匹配的字符。如果执行成功，则返回剩余字符串（存在相匹配的字符）；如果没有找到相匹配的字符，则返回 FALSE。

 **说明：** 有关文件上传技术的讲解请参考本章实例 227，这里不再赘述。

设计过程

（1）创建 index.php 文件。

（2）添加表单，设置文件域、提交按钮，使用 POST 方法，设置 enctype="multipart/form-data"，将数据提交到本页。

（3）通过$_FILES 获取上传文件的相关信息。

（4）应用 is_dir()函数判断指定的服务器文件夹是否存在，如不存在则应用 mkdir()函数创建文件夹。

（5）应用 is_uploaded_file()函数判断文件是否是 HTPP POST 上传。

（6）获取上传文件名称的后缀名，根据后缀名的不同，定义不同的存储路径。

（7）应用 move_uploaded_file()函数执行文件上传的操作。

index.php 文件的关键代码如下：

```
<?php
if (! empty ( $_FILES [up_picture] [name] )) {                    //判断上传内容是否为空
 if ($_FILES ['up_picture'] ['error'] > 0) {                       //判断文件是否可以上传到服务器
        echo "上传错误:";
        switch ($_FILES ['up_picture'] ['error']) {
                case 1 :
                        echo "上传文件大小超出配置文件规定值";
                        break;
                case 2 :
                        echo "上传文件大小超出表单中约定值";
                        break;
                case 3 :
                        echo "上传文件不全";
                        break;
                case 4 :
                        echo "没有上传文件";
                        break;
        }
 }
} else {
 if (! is_dir ( "./txt/" )) {                                      //判断指定目录是否存在
        mkdir ( "./txt/" );                                        //创建目录
 }
 if (! is_dir ( "./pic/" )) {                                      //判断指定目录是否存在
        mkdir ( "./pic/" );                                        //创建目录
 }
 if (! is_dir ( "./fla/" )) {                                      //判断指定目录是否存在
        mkdir ( "./fla/" );                                        //创建目录
 }
}
if (is_uploaded_file ( $_FILES ['up_picture'] ['tmp_name'] )) {   //判断文件是否是 HTPP POST 上传
 $type = $_FILES ['up_picture'] ['name'];                          //获取上传文件的名称
 $types = strtolower ( strstr ( $type, '.' ) );                    //获取上传文件的后缀
 if ($types == ".txt" || $styles == ".doc") {
        $path = './txt/' . time () . strstr ( $_FILES ['up_picture'] ['name'], '.' );    //定义上传文件名称和存储位置
 } elseif ($types == ".jpg" || $styles == ".gif" || $styles == ".bmp") {
        $path = './pic/' . time () . strstr ( $_FILES ['up_picture'] ['name'], '.' );    //定义上传文件名称和存储位置
 } else {
        $path = './fla/' . time () . strstr ( $_FILES ['up_picture'] ['name'], '.' );    //定义上传文件名称和存储位置
 }
 if (! move_uploaded_file ( $_FILES ['up_picture'] ['tmp_name'], $path )) {    //执行上传操作
        echo "上传失败！";
 } else {
        echo "文件类型：" . $types . " 上传成功，大小为：" . $_FILES ['up_picture'] ['size'];
 }
} else {
 echo "上传文件：" . $_FILES ['up_pictute'] ['name'] . "不合法！";
}
?>
```

秘笈心法

心法领悟 236：文件类型检测的妙用。

文件后缀名的获取可以应用到很多地方，最为常用的就是通过它来控制上传文件的类型，另外在文件的读取中，用于判断哪些文件可以读取，哪些不可以。

实例 237 判断文件的权限

光盘位置：光盘\MR\04\237

初级

趣味指数：★★☆

实例说明

文件权限的判断是一个非常重要的功能，如果文件没有读、写的权限，那么就不能对其进行任何操作。本实例将介绍如何判断文件的权限。运行本实例，通过文本框提交文件路径，向指定的文件中写入数据，如果文件具备写的权限，则可以实现数据的写入，否则将提示文件不具备写入的权限，运行结果如图 4.15 所示。



图 4.15　判断文件的权限

关键技术

判断文件是否具备读权限应用的是 is_readable()函数，判断文件是否具备写权限应用的是 is_writable()函数。

is_readable()函数，判断指定文件是否可读，语法如下：

```
bool is_readable ( string filename )
```

如果文件存在并且可读则返回 TRUE。

is_writable()函数，判断指定文件是否可写，语法如下：

```
bool is_writable ( string filename )
```

如果文件存在并且可写则返回 TRUE。参数 filename 可以是一个允许进行是否可写检查的目录名。

注意：PHP 也许只能以运行 webserver 的用户名（通常为 'nobody'）来访问文件。不计入安全模式的限制。

对于文件权限的判断，是对文件进行操作的前提，特别是在执行文件的读取、写入、重命名等操作时，如果文件不具备读、写的权限，那么这些操作是没有任何意义的。

设计过程

（1）创建 index.php 文件。

（2）添加表单，设置文本框，提交指定文件，设置文件域，提交写入文件中的数据，设置提交按钮，使用 POST 方法，将数据提交到本页。

（3）通过$_POST[]方法获取表单提交的文件路径和文件内容，并通过 iconv()函数对获取的数据进行编码转换。首先，判断指定的文件是否存在。然后，判断指定的文件是否具备写的权限，如果具备，则将表单提交的文件内容写入到文件中。最后，关闭打开的文件，给出提示信息。index.php 的关键代码如下：

```
<?php
	if ($_POST ['file_name'] != "" && is_file ( iconv ( "utf-8", "gb2312", $_POST ['file_name'] ) ) == true) {	//判断文件是否存在
		$file_name = iconv ( "utf-8", "gb2312", $_POST ['file_name'] );	//编码转换
		$file_content = iconv ( "utf-8", "gb2312", $_POST ['file_content'] );
		if (file_exists ( $file_name )) {
```

```
                if (is_writable ( $file_name )) {                              //判断文件是否具备写的权限
                        $fp = fopen ( $file_name, "w+" );                      //打开指定的文件
                        if (fwrite ( $fp, $file_content )) {                   //执行写入的操作
                                echo "<script>alert('文件写入成功!');</script>";
                        } else {
                                echo "<script>alert('文件写入失败!');</script>";
                        }
                        fclose ( $fp );                                        //关闭文件
                } else if (is_readable ( $file_name )) {                       //判断文件是否具备读的权限
                        echo "<script>alert('文件只具备读权限!');</script>";
                } else {
                        echo "<script>alert('文件不具备读、写权限!');</script>";
                }
        } else {
                echo "<script>alert('文件不存在！ ');</script>";
        }
} else {
        echo "<script>alert('请输入正确的文件路径！ ');</script>";
}
?>
```

秘笈心法

心法领悟 237：通过 fileperms()函数取得文件的权限。

本实例中介绍的是通过 is_readable()和 is_writable()函数来判断文件是否具有读、写的权限。在文件系统的函数中还可以通过 fileperms()函数取得文件的权限，其返回值是一个 int 型的字符串。

实例 238　从文本文件中读取注册服务条款

光盘位置：光盘\MR\04\238

高级

趣味指数：★★★★☆

实例说明

在网站开发的过程中，经常会创建注册服务条款或者会员须知之类的文件。处理此类文件最直接的方法是将其生成一个独立的页面，而最实用的方法是将其存储于独立的文本文件中，从文本文件中读取这些服务条款，这样不占用数据库的空间，而且不占用过多的页面。运行本实例，实现一个用户注册的功能，并从文本文件中读取服务条款，运行结果如图 4.16 所示。

图 4.16　从文本文件中读取注册服务条款

关键技术

从文本文件中读取注册服务器条款，可以分为两个部分：第一部分是表单中文件域的创建；第二部分是应用文件系统函数读取、输出指定文本文件中的数据。

这里应用文件系统中的 file()函数读取文本文件中的数据，有关文本文件中数据的读取可以参考 4.3 节中的内容，这里不再赘述。

设计过程

（1）创建 index.php 文件。

首先，设计用户注册页面。然后，添加用户注册的表单及表单元素，将表单元素的值提交到 index_ok.php 页。接着，编辑 JavaScript 脚本，通过 chkreginfo()方法对表单元素的值进行判断。

最后，在文本域中编写 PHP 脚本，通过 file()函数读取指定文本文件中的数据，通过 foreach()语句循环输出注册服务条款，其关键代码如下：

```
<textarea name="register" cols="40" rows="5" id="register">
<?php
	$arr = file ( 'files/register.txt' );
	foreach ( $arr as $value ) {
		echo iconv ( "gb2312", "utf-8", $value );
	}
?>
</textarea>
```

（2）创建 index_ok.php 文件，通过$_POST[]方法获取表单中提交的数据，输出用户的注册信息。

（3）在根目录下创建 js 脚本文件夹，编写 check.js 脚本文件，完成对表单元素值的判断。

秘笈心法

心法领悟 238：读取文本文件中数据时函数的选择。

在文件系统函数中，有多个可以读取文本文件中数据的函数，为了使读取到的数据输出时条理更加清晰，建议使用 file()函数，因为该函数将读取到的数据写入一个数组中，该数组中的每个元素是文件中对应的一行，这样在通过 foreah()语句输出数组中元素时，是按照原文件中的行进行输出的，确保了内容输出的条理性。

实例 239	可以屏蔽刷新功能的文本计数器 光盘位置：光盘\MR\04\239	高级 趣味指数：★★★★☆

实例说明

网站的计数器对于网站管理者来说是一个非常值得关注的部分，它记录了网站被访问的次数，客观地反映了网站受欢迎的程度。而文本计数器是最简单的一种，它将数据存储于文本文件中。在本实例中，将介绍这种文本计数器的实现方法，并且将重点阐述如何屏蔽网页刷新对计数器的影响。运行本实例，将输出如图 4.17 所示的内容。

最新动态 · 此时是：2010年7月6日16:33:51　您是本网站第1800008 位访客！

图 4.17　可以屏蔽刷新功能的文本计数器

此时，无论如何刷新当前页面，计数器统计的值都不会再发生变化，除非关闭此页面，重新打开。

关键技术

文本计数器的设计思路如下：首先，判断文本文件是否存在，如不存在则打开失败；如打开成功则继续执行并读取文件中的数据，将计数器增加 1。然后，以写的方式重新打开文件，把新的统计数据写入文件后，关闭文件。最后，重新打开文件，读取并输出文件中的数据。

操作流程如图 4.18 所示。

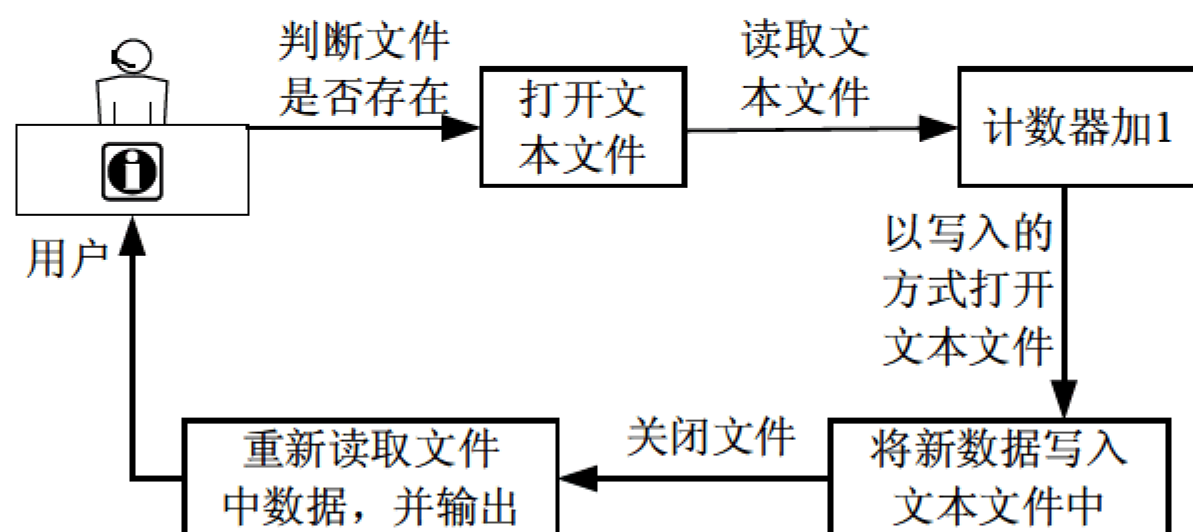


图 4.18　文本计数器的操作流程

按照这个原理设计的计数器，当刷新页面时计数器的值也会增加，那么这个计数器就没有任何意义。所以要想这个计数器有意义，必须屏蔽页面刷新对计数器的影响。

这里通过 SESSION 来实现这个功能，其原理如图 4.19 所示。

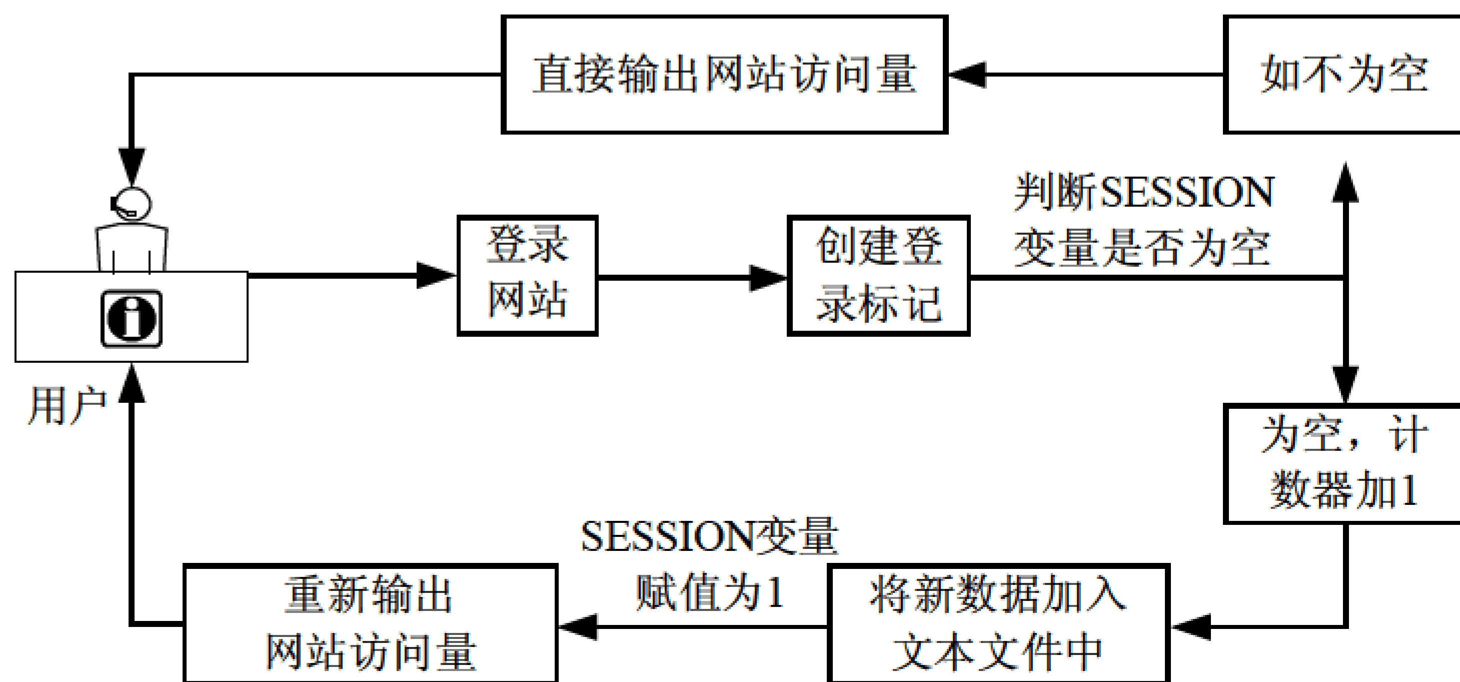


图 4.19　网站计数器的设计原理

设计过程

（1）创建 index.php 文件。首先，设计网页页面。然后，初始化 SESSION 变量。接着，编写 PHP 脚本，通过文件系统函数完成网站访问量的统计，通过 SESSION 屏蔽网页刷新对计数器的影响。最后，输出网站的访问量和当前时间，其关键代码如下：

```
<?php
session_start ();                          //初始化一个 session 变量
?>
<?php
//使用文本存储数据
if ($_SESSION [temp] == "") {              //判断$_SESSION[temp]==""的值是否为空，其中的 temp 为自定义的变量
 if (($fp = fopen ( "counter.txt", "r" )) == false) {
        echo "打开文件失败!";
 } else {
        $counter = fgets ( $fp, 1024 );    //读取文件中数据
        fclose ( $fp );                    //关闭文本文件
        $counter ++;                       //计数器增加 1
```

```
        $fp = fopen ( "counter.txt", "w" );                  //以写的方式打开文本文件
        fputs ( $fp, $counter );                              //将新的统计数据增加 1
        fclose ( $fp );                                       //关闭文件
    }
    $_SESSION [temp] = 1;                                     //计数器的值增加后，为$_SESSION[temp]赋值 1
}
//从文本文件中读取统计数据
if (($fp = fopen ( "counter.txt", "r" )) == false) {
        echo "打开文件失败!";
} else {
        $counter = fgets ( $fp, 1024 );                       //读取文本文件的数据
        fclose ( $fp );
}
?>
```

（2）在实例根目录下创建 counter.txt 文本文件，用于存储网站的访问量。

秘笈心法

心法领悟 239：SESSION 防止重复计数的原理解析。

首先，在网页被访问时，初始化一个 SESSION 变量，并赋给其一个空值。然后，判断 SESSION 变量的值是否为空，如果为空，则将计数器的值增加 1，并且为 SESSION 变量赋值为 1。此时，在当前页中，SESSION 变量的值已经不为空，无论如何刷新，SESSION 变量的值都不会改变，所以计数器的值也不会增加。

实例 240　判断文件是否被修改

光盘位置：光盘\MR\04\240

初级　趣味指数：★★★☆

实例说明

在网站的管理系统中，有时需要查看某个文件是否被修改过、在什么时间被修改的、最后的修改时间是什么时候，本实例就可以实现这个功能，对表单中提交的文件进行判断，检测出修改时间，其运行结果如图 4.20 所示。



图 4.20　检测文件是否被修改

关键技术

本实例主要应用 filectime()和 filemtime()函数，检测文件的 incode 最后改变时间和最后的修改时间，并应用 date()函数对检测返回的时间戳进行格式化。

filectime()函数，返回指定文件 filename 的 inode 最后改变时间，语法如下：

```
int filectime(string filename);
```

成功则返回 UNIX 时间戳，否则返回 FALSE。

filemtime()函数，返回指定文件 filename 的最后修改时间，语法如下：

```
int filemtime(string filename);
```

成功则返回 UNIX 时间戳，否则返回 FALSE。

设计过程

首先，设计网页页面。然后，创建一个表单，通过文件域提交要判断的文件。接着，获取表单中提交的文件路径，应用 filectime()和 filemtime()函数对提交的文件进行检测。最后，输出检测结果。

index.php 文件的关键代码如下：

```
<?php
$file = iconv ( "utf-8", "gb2312", $_POST ['files'] );                //实现编码格式的转换
if (file_exists ( $file )) {                                           //判断文件是否存在
        $change_time = filectime ( $file );                            //获取文件的最后 incode 时间
        $time = date ( "Y-m-d h:i:s", $change_time );                  //时间戳的格式化
        $last_time = filemtime ( $file );                              //获取文件的最后修改时间
        $times = date ( "Y-m-d h:i:s", $last_time );                   //时间戳的格式化
} else {
        $result = "该文件不存在!!";
}
?>
```

秘笈心法

心法领悟 240：获取文件的最后访问时间。

通过文件系统函数不但可以获取文件的最后修改时间，而且可以获取文件的最后访问时间，其应用的是 fileatime()函数。

实例 241	重新定义上传文件的名称 光盘位置：光盘\MR\04\241	高级 趣味指数：★★★★☆

实例说明

应用 POST 方法上传文件时，需要将上传文件保存到服务器指定的目录中，这时可能会出现因名称相同而文件相互替换的情况。为了解决上述问题，可以应用 basename()函数和随机函数 mt_rand()对上传文件进行重新命名。在本实例中，成功上传一个文件之后，将弹出新的文件名称对话框，如图 4.21 所示。



图 4.21　重新定义上传文件名称

关键技术

在本实例中，实现文件上传的功能，有关文件上传的技术请参考本章实例 224，这里不再赘述。下面将介绍一种给上传文件命名的方法。

basename()函数，返回指定文件目录中的基本文件名，语法如下：

```
string basename(string path [, string suffix])
```

参数 path 指定文件的路径；参数 suffix 为可选参数，如果文件路径以 suffix 结尾，那么这部分内容将被删除。

通过此函数获取到上传文件的原始名称，并对这个名称进行重新定义，进而避免在将文件上传到服务器时出现重名的问题。

设计过程

上传文件的具体实现步骤可以参考本章实例 224，这里不再赘述。

在本实例中，关键是在定义上传文件的名称时，应用随机函数 mt_rand()、时间戳以及 basename()函数，其关键代码如下：

```
$date=date("YmdHis");                                                    //定义随机数
$filename=mt_rand(1000,9999).$date.basename($_FILES['up_picture']['name']);   //定义上传文件名称
echo "<script>alert('文件名被替换为".$filename."');</script>";
$path='./upfile/'.$filename;                                             //定义上传文件名称和存储位置
```

秘笈心法

心法领悟 241：上传文件原始名称的获取。

在通过表单中的文件域提交上传文件时，如果在 form 中定义了 enctype="multipart/form-data"属性，那么在获取上传文件的原始名称时必须使用$_FILES 全局变量，如果使用$_POST 是获取不到值的。

实例 242 读取远程文件中的数据

光盘位置：光盘\MR\04\242

高级

趣味指数：★★★★☆

实例说明

设想这样一个场景，若知道某个网站中一个文件的具体路径，我们在本地就可以完成对这个文件内容的读取操作，这将是一件多么有趣的事情。本实例就将这个场景变为现实，实现读取远程文件的功能，运行结果如图 4.22 所示。



图 4.22　读取远程文件的数据

 **说明：**此功能的实现会有诸多的阻碍，例如，服务器不支持文件的远程操作、访问的文件不具备读的权限等。这里只是通过这样一个实例，告诉大家如何实现远程文件的访问操作。如果服务器支持文件的远程访问，那么会有一定的安全隐患。

关键技术

访问远程文件应用 fopen()函数，只要将其参数 filename 的值设置为一个完整的 URL 地址即可。

 **技巧：**在 PHP 中，如果要访问远程文件，必须将配置文件 php.ini 中的参数 allow_url_fopen 设置为开启。allow_url_fopen 参数默认是开启的，允许打开 http 协议和 ftp 协议指定的远程文件，如果 allow_url_fopen 设置为 off，则不允许打开远程文件，函数将返回 FALSE。

设计过程

（1）创建 index.php 文件。首先，设计网页页面。然后，创建一个表单，通过文本框提交要访问的远程文件的地址。最后，获取表单中提交的远程文件路径，并通过 header()函数跳转到 read.php 文件中，在该文件中完成对远程文件的读取操作。index.php 文件的关键代码如下：

```
<?php
if($_POST ['files']!=""){
 $filenames=urlencode($_POST ['files']);
 header("Location: read.php?filename=$filenames");
}
?>
```

（2）创建 read.php 文件，获取 header()函数传递的参数值，应用 file()函数读取参数值中传递的远程文件，并通过 foreach()语句输出远程文件的内容，其代码如下：

```
<?php
$filename = iconv ( "utf-8", "gb2312", $_GET['filename'] );          //获取 header()函数传递的参数值
$arr = file ( $filename ) ;                                           //读取远程文件的内容
foreach ( $arr as $value ) {                                          //循环输出数据
 echo htmlspecialchars ( iconv ( "gb2312", "utf-8", $value ) )."<br>";
}
?>
```

秘笈心法

心法领悟 242：不被看好的远程文件访问功能。

严格地说，远程文件的访问功能在生产环境是不被支持的，因为它涉及到服务器中文件的安全问题，所以在大多数的服务器中，allow_url_fopen 的值都是关闭的。

实例 243　删除指定目录下的所有.ini 文件

光盘位置：光盘\MR\04\243

高级

趣味指数：★★★★☆

实例说明

在本实例中，将完成一个具体的操作——删除指定目录下的所有.ini 文件，其运行结果如图 4.23 所示。

当前目录:F:\AppServ\www\MR\04\023

项目名	大小	创建日期	最后修改时间	操作
锁定	目录	2010-07-07 03:12:19	2010-07-07 03:12:20	删除
上级目录	目录	2010-06-30 01:05:48	2010-06-30 01:05:50	删除
index.php	5116	2010-07-07 03:12:19	2010-07-07 03:28:14	index.php
delete.php	713	2010-07-07 03:12:19	2010-07-01 09:06:28	delete.php
look_file.php	876	2010-07-07 03:12:19	2010-07-07 03:25:46	look_file.php
IMAGES	目录	2010-07-07 03:12:19	2010-07-07 03:12:20	IMAGES
配置文件.ini	0	2010-07-07 03:26:43	2010-07-07 03:26:44	删除
php.ini	0	2010-07-07 03:26:52	2010-07-07 03:26:44	删除

All CopyRights © reserved 2010 吉林省明日科技有限公司

图 4.23　删除指定目录下的所有.ini 文件

关键技术

有关目录、文件的删除方法已经在前面的实例中讲解过，这里就不再赘述。删除指定目录下特定格式文件的实现原理与删除所有文件相同，只是在创建删除链接前对文件的格式进行判断，如果文件是.int 类型，那么就

输出删除的超链接，否则将直接输出文件的名称，其关键代码如下：

```
$catalog = getcwd () . "\\$gain_directory";                       //子目录
        $ext = substr ( $gain_directory, strrpos ( $gain_directory, "." ) );     //获取文件的后缀
        if (strtoupper ( $ext ) == ".INI") {                          //判断如果文件的后缀是.ini，则创建删除超链接
                echo "<a href='delete.php?catalog=".urlencode($catalog)."&filename=" . urlencode(getcwd ()) . "' title='删除目录或者文件' >删除</a>";
        } else {                                                      //如果文件的后缀不是.ini，则直接输出文件名称
                echo iconv ( "gb2312", "utf-8", $gain_directory );
        }
```

设计过程

（1）本实例 index.php 文件的创建过程与遍历、删除指定目录下的所有文件是基本相同的，不同之处已经在关键技术中给出，这里不再赘述。

（2）创建 look_file.php 文件，完成对.ini 后缀文件的读取操作，其代码如下：

```
<?php
$catalog = iconv ( "utf-8", "gb2312", urldecode ( $_GET [catalog] ) );
$filename = iconv ( "utf-8", "gb2312", urldecode ( $_GET [filename] ) ); $type = iconv ( "utf-8", "gb2312", urldecode ( $_GET [type] ) );   //获取文件的信息,设置编码
$arr = file ( $catalog . "\\" . $filename );                    //读取文件
foreach ( $arr as $value ) {                                    //循环输出文件内容
 $value = htmlentities ( $value, ENT_COMPAT, "UTF-8" );         //特殊字符的转换
 echo $value . "<br>";                                          //输出内容
}
?>
```

（3）创建 delete.php 文件，完成对指定文件的删除操作，代码请参考本书光盘。

秘笈心法

心法领悟 243：合理地运用自己的编程资源。

在进行程序开发的过程中，要学会合理地运用自己的编程资源。一个好的功能模块，不只是在一个程序中可以使用还应具有重用价值。诸如数据库的连接、操作方法以及分页技术等，一次开发后将会在很多的程序中使用，避免对类似程序进行重复开发，浪费不必要的时间。

作为一个程序员，不只要每次在开发时都考虑新的功能、新的技术，更重要的是充分运用已有资源，既可以提高程序的开发效率，又可以避免开发新技术产生的一些问题。

实例 244 将文本文件中数据存储到数据库中

光盘位置：光盘\MR\04\244

高级

趣味指数：★★★★☆

实例说明

文本文件中的数据也可以转存到数据库中。例如，在编程词典服务网中，有一个编程词典系列软件注册信息提交页面，在该页面中，编程词典用户不但提交个人信息，而且将安装编程词典生成的注册信息文件提交到服务器中，在提交注册信息的同时，将注册信息中的数据与用户个人信息一起存储到数据库中。本实例模拟这个功能，开发一个将文本文件上传到服务器，并且将文本文件中数据转存到数据库中的实例，其运行结果如图 4.24 所示。

关键技术

本实例运用了三方面的技术：第一，将文本文件上传到服务器；第二，读取服务器指定文件夹下文本文件中的数据；第三，连接数据库，将从文本文件中读取的数据存储到指定数据表中。



图 4.24　将文本文件存储到数据库中

其中，最为关键的是将文本文件中的数据存储到数据库中，在执行数据的读取和存储时，有两个细节必须把握。第一个，应用 addslashes()函数，对从文本文件中读取的数据进行转义，因为在读取的数据中可能包含单引号（'）、双引号（"）、反斜线（\）或者 NUL（NULL 字符），如果不对其进行转换，在执行 SQL 语句时可能会出现错误。

addslashes()函数，通过反斜线来引用字符串，语法如下：

```
string addslashes ( string str)
```

在其返回的字符串中，为了数据库查询语句等的需要在某些特定字符前加上了反斜线，这些特定字符包括单引号（'）、双引号（"）、反斜线（\）和 NUL（NULL 字符）。

说明： 默认情况下，PHP 指令会自动对所有的 GET、POST 和 COOKIE 数据运行 addslashes()函数。所以，如果是通过 GET 或者 POST 方法提交的数据是不需要进行转义的，可以直接添加到数据库中；如果提交的数据是从文本文件中读取或者自定义的内容，那么就有必要对其进行转义。

设计过程

（1）创建 index.php 文件。首先，设计网页页面。然后，创建一个表单，通过文件域提交要上传的.ini 文件。接着，获取表单中提交的文件，将文件存储到服务器指定的文件夹下，通过 file_get_contents()函数读取上传文件的内容，并且将读取的结果存储到指定的数据表中。最后，执行查询语句，输出数据库中存储的.ini 文件的内容，其关键代码如下：

```
<?php
if (! empty ( $_FILES [up_picture] [name] )) {                    //判断上传内容是否为空
 $type = $_FILES ['up_picture'] ['name'];
 $types = strstr ( $type, '.' );                                  //获取文件后缀
 if ($types == ".ini") {
        if ($_FILES ['up_picture'] ['error'] > 0) {               //判断文件是否可以上传到服务器
                echo "上传错误:";
                switch ($_FILES ['up_picture'] ['error']) {
                        case 1 :
                                echo "上传文件大小超出配置文件规定值";
                                break;
                        case 2 :
                                echo "上传文件大小超出表单中约定值";
                                break;
                        case 3 :
                                echo "上传文件不全";
                                break;
                        case 4 :
                                echo "没有上传文件";
```

```
                    break;
                }
        } else {
                if (! is_dir ( "./upfile/" )) {                                  //判断指定目录是否存在
                        mkdir ( "./upfile/" );                                   //创建目录
                }
                $file_name = time () . strstr ( $_FILES ['up_picture'] ['name'], '.' );        //定义上传文件名称
                $path = './upfile/' . $file_name;                                 //定义上传文件名称和存储位置
                if (is_uploaded_file ( $_FILES ['up_picture'] ['tmp_name'] )) {  //判断文件是否是 HTPP POST 上传
                        if (! move_uploaded_file ( $_FILES ['up_picture'] ['tmp_name'], $path )) { //执行上传操作
                                echo "上传失败";
                        } else {
                                //读取文本文件中数据，并且实现转义和编码的转换
                                $arr = iconv ( "gb2312", "utf-8", addslashes ( file_get_contents ( $path ) ) );
                                include_once ("conn/conn.php");                   //连接数据库
                                $sql = "insert into tb_files(file_name,file_content,file_date)values('$file_name','$arr','" . date ( "Y-m-d H:i:s" ) . "')";
                                                                                  //定义 SQL 添加语句
                                $result = mysql_query ( $sql, $conn );            //执行添加操作
                                echo "文件：  " . $_FILES ['up_picture'] ['name'] . "  上传成功!" . "<br>";
                                echo "大小：  " . $_FILES ['up_picture'] ['size'] . "  字节";
                        }
                } else {
                        echo "上传文件" . $_FILES ['up_pictute'] ['name'] . "不合法！ ";
                }
        }
} else {
        echo "上传文件" . $_FILES ['up_pictute'] ['name'] . "类型不正确！ ";
}
}
?>
```

（2）创建 conn 文件夹，编写 conn.php 文件，连接数据库服务器，连接 db_database04 数据库，其代码如下：

```
<?php
$conn = mysql_connect ( "localhost", "root", "111" ) or die ( "连接数据库服务器失败！ " . mysql_error () );
mysql_select_db ( "db_database04", $conn );                              //选择数据库 db_database04
mysql_query ( "set names utf8" );                                        //设置数据库编码格式 utf-8
?>
```

秘笈心法

心法领悟 244：addslashes()转义函数的合理运用。

将从文本文件中读取的数据存储到数据库时，必须应用 addslashes()函数对文本文件返回的字符串进行转义，否则有可能出现如图 4.25 所示的错误。

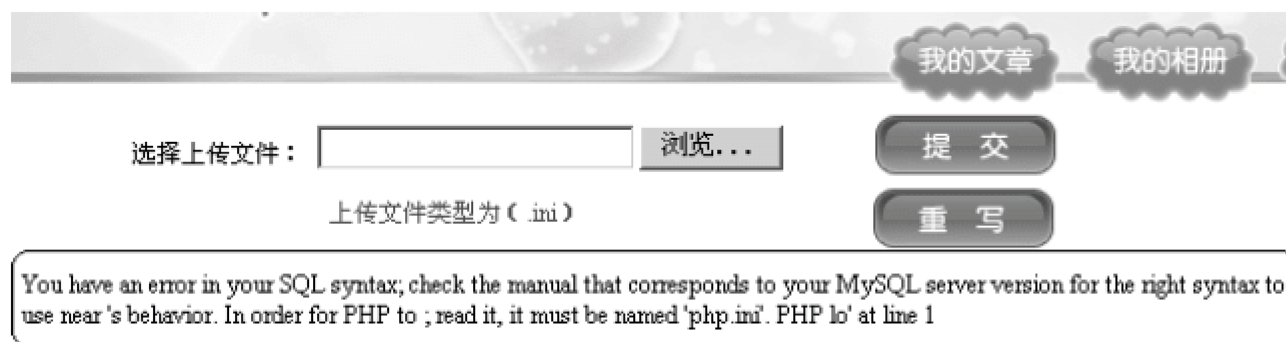


图 4.25　数据添加失败

4.5　文件的压缩与解压

WinRAR 是一个非常实用的文件压缩管理工具，利用其强大的功能可以很方便地将多个文件压缩成为一个文件，这样不仅可以方便操作，而且可以在很大程度上节省文件存储的空间。本节将向读者介绍如何利用 PHP 解压缩和解压缩文件。

实例 245　PHP 中压缩 RAR 文件

光盘位置：光盘\MR\04\245

高级

趣味指数：★★★★★

实例说明

在网站中，上传的文件大多数都是以压缩包的形式存在的，所以先将要上传的文件压缩后才能够上传，如果能在网站中增加上传文件自动压缩模块，将会给用户带来很大的方便。本实例将向读者介绍如何在 PHP 中完成文件的压缩操作。运行本实例，首先选择要压缩的文件，然后设置压缩文件的存储位置，单击“提交”按钮，运行结果如图 4.26 所示。



图 4.26　RAR 压缩文件

关键技术

文件压缩主要应用 PHP 的预定义类 com，将 wscript.shell 对象作为 com 类的参数，最终通过调用 com 类的 run()方法实现文件的压缩。

com 类，提供一个将 COM 组件整合到 PHP 脚本中的框架，实例化方法如下：

```
$obj = new COM("server.object")
```

COM 类构造函数的语法如下：

```
string COM::COM ( string module_name [, string server_name [, int codepage]] )
```

参数说明：

module_name：被请求组件的名字或 class-id。

server_name：DCOM 服务器的名字，组件在此服务器上被取用。如果是 NULL，则假定是 localhost。如果要支持允许 DCOM，必须将 php.ini 中的 com.allow_dcom 设为 TRUE。

codepage：指定用于将 PHP 字符串（php-strings）转换成 UNICODE 字符串（unicode-strings）的代码页，反之亦然。可用的值为 CP_ACP、CP_MACCP、CP_OEMCP、CP_SYMBOL、CP_THREAD_ACP、CP_UTF7 和 CP_UTF8。

com 类中 run()方法的语法如下：

```
run("winrar a  生成的 rar 文件的路径及名称  要压缩的文件的路径及名称",1,false);
```

设计过程

（1）创建 index.php 文件。首先，创建一个表单，设置文件域，提交要压缩的文件；设置文本框，提交压缩文件的存储位置；设置按钮，完成表单元素的提交操作。

（2）在 index.php 文件中编写 PHP 脚本，通过$_POST[]方法获取表单提交的数据，通过 iconv()函数完成数据编码格式的转换，实例化 com 类，调用 run()方法完成文件的压缩操作，其关键代码如下：

```
<?php
if ($_POST ['Submit'] != "") {
        $sourcefile = iconv ( "utf-8", "gb2312", $_POST ['sourcefile'] );              //获取指定文件的路径
        $objfile = $_POST ['objfile'] . "\\" . time ();                                //定义压缩文件的存储位置和名称
```

```
        $objfile = iconv ( "utf-8", "gb2312", $objfile );                    //实现编码转换
        $obj = new com ( "wscript.shell" );                                  //实例化 com 类
        $obj->run ( "winrar a " . $objfile . " " . $sourcefile . "", 1, false );  //调用类中的方法
        echo "<script>alert('文件压缩完成!');</script>";
}
?>
```

秘笈心法

心法领悟 245：执行文件压缩操作的误区。

在执行文件压缩操作时，切忌在文件的存储路径中出现空格（包括被压缩文件的路径、名称以及压缩文件的存储路径），否则压缩操作将不会正确执行。

实例 246　PHP 中将上传文件转换成 RAR 文件　高级

光盘位置：光盘\MR\04\246　趣味指数：★★★★☆

实例说明

在本实例中，实现将上传的文件直接以压缩文件的格式进行存储，其好处是可以缓解服务器的压力、节省服务器的空间。运行本实例，选择要上传的文件，单击“提交”按钮，运行结果如图 4.27 所示。



图 4.27　将上传文件转换成压缩文件

上传成功后，可以查看本实例根目录下的 upfile 文件夹，确定上传文件是否以压缩文件的格式进行存储。

关键技术

本实例实现文件上传技术与文件压缩技术的结合，进而实现将上传文件以 RAR 格式进行存储。其中在调用 com 类中的 run()方法执行压缩操作时，在 winrar 的命令中，增加了两个属性：“-ep”将压缩文件以文件名的格式进行存储，不包括文件的路径；“-df”文件压缩成功后删除原始文件，其完整的命令语句如下：

```
$obj->run ( "winrar a -ep -df " . $file_path.".rar" . " " . $path . "", 1, false );  //调用类中的方法，执行文件压缩操作
```

设计过程

（1）创建 index.php 文件。首先，创建一个表单，设置文件域，提交上传文件；设置按钮，完成表单元素的提交操作。

（2）在 index.php 文件中编写 PHP 脚本。首先，对上传文件进行判断，如果符合要求将上传文件存储到服务器指定文件夹下。然后，实例化 PHP 中的 com 类，通过 run()方法执行文件的压缩命令，完成上传文件的压缩操作，并将原始文件删除，其关键代码如下：

```
//定义上传文件名称和存储位置
$path = './upfile/' . time () . strstr ( $_FILES ['sourcefile'] ['name'], '.' );
 if (is_uploaded_file ( $_FILES ['sourcefile'] ['tmp_name'] )) {
 if (! move_uploaded_file ( $_FILES ['sourcefile'] ['tmp_name'], $path )) {                echo "上传失败";
        } else {
```

```
            $path = realpath ( iconv ( "utf-8", "gb2312", $path ) );          //获取文件在服务器中的路径
            $len = strripos ( $path, "." );                                   //计算路径的长度
            $file_path = substr ( $path, 0, $len );                           //获取上传文件的路径名称
            $obj = new com ( "wscript.shell" );                               //实例化 com 类
//调用类中的方法，执行文件压缩和原始文件删除操作
            $obj->run ( "winrar a -ep -df " . $file_path . ".rar" . " " . $path . "", 1, false );
            echo "文件：" . $_FILES ['sourcefile'] ['name'] . "上传成功！" . "<br>";
            echo "大小：" . $_FILES ['sourcefile'] ['size'] . " 字节";
        }
} else {
        echo "上传文件" . $_FILES ['sourcefile'] ['name'] . "不合法！";
}
```

秘笈心法

心法领悟 246：winrar 操作命令的运用。

对 RAR 文件进行操作的命令，不只 r、-ep 和-df 这 3 个，还有很多，读者可以参考 RAR 用户手册了解更多关于 RAR 操作的命令。

实例 247　PHP 中对 RAR 文件进行解压

光盘位置：光盘\MR\04\247

高级

趣味指数：★★★★☆

实例说明

既然已经介绍了如何对文件进行压缩，那么必然要介绍如何对压缩文件进行解压。运行本实例，首先选择要解压的 RAR 文件，并在文本框中输入要解压文件的存储目录，单击“解压”按钮即可，运行结果如图 4.28 所示。



图 4.28　解压 RAR 文件

关键技术

文件解压同样应用 PHP 的预定义类 com，将 wscript.shell 对象作为 com 类的参数，通过调用 com 类的 run()方法执行文件解压的命令。文件解压的命令如下：

```
$obj->run ( "winrar x -av -c ".$sourcefile." ".$objfile. "", 1, true );     //调用类中的方法，执行文件解压的命令
```

通过此命令将指定的压缩文件解压到指定的绝对路径下。

设计过程

（1）创建 index.php 文件。首先，创建一个表单，设置文件域，提交要解压的文件；设置文本框，提交解压后文件的存储位置；设置按钮，完成表单元素的提交操作。

（2）在 index.php 文件中编写 PHP 脚本，通过$_POST[]方法获取表单提交的数据，通过 iconv()函数完成数据编码格式的转换，实例化 com 类，调用 run()方法执行文件的解压命令，其关键代码如下：

```
<?php
if ($_POST ['Submit'] != "") {
```

```
        include("function.php");
        $sourcefile = iconv ( "utf-8", "gb2312", unhtml($_POST ['sourcefile']) );      //获取指定文件的路径
        $objfile = iconv ( "utf-8", "gb2312", $_POST ['objfile'] );                   //定义解压文件的存储位置和名称
        $obj = new com ( "wscript.shell" );                                          //实例化 com 类
        $obj->run ( "winrar x -av -c ".$sourcefile." ".$objfile. "", 1, true );       //调用类中的方法，执行解压命令
        echo "<script>alert('文件解压完成!');</script>";
}
?>
```

秘笈心法

心法领悟 247：执行文件解压操作的误区。

在执行文件解压操作时，切忌在文件的存储路径中出现空格（包括被解压文件的路径、名称以及解压后文件的存储路径），否则解压操作将不会正确执行。

实例 248　PHP 中压缩 ZIP 文件

光盘位置：光盘\MR\04\248

高级

趣味指数：★★★★☆

实例说明

通过 PHP 不但可以将文件压缩成 RAR 格式，而且可以将文件压缩成 ZIP 格式，本实例将介绍如何应用 PHP 将文件压缩成 ZIP 格式。运行本实例，首先选择要压缩的文件，然后设置压缩文件的存储位置，单击“提交”按钮即可，运行结果如图 4.29 所示。



图 4.29　ZIP 压缩文件

关键技术

运用 PHP 将文件压缩成 ZIP 格式的原理与压缩成 RAR 格式的原理相同：首先，运用 PHP 的预定义类 com 调用系统中安装 WinRAR 后所形成的组件 wscript.shell，并对 com 类进行实例化。然后，调用 com 类中的 run()方法执行文件的压缩命令。对 com 类进行实例化及调用类中 run()方法的代码如下：

```
$obj=new com("wscript.shell");
$obj->run("winrar a ".$objfile." ".$sourcefile."",1,false);。
```

设计过程

（1）创建 index.php 文件。编写 form 表单，提交被压缩文件和压缩文件的存储位置及名称。

（2）在 index.php 文件中，编写 PHP 脚本，通过$_POST[]方法获取表单提交的数据，通过 iconv()函数完成数据编码格式的转换，实例化 com 类，调用 run()方法完成文件的压缩操作，其关键代码如下：

```
<?php
if ($_POST ['Submit'] != "") {
        $sourcefile = iconv ( "utf-8", "gb2312", $_POST ['sourcefile'] );      //获取指定文件的路径
        $objfile = iconv ( "utf-8", "gb2312", $_POST ['objfile'] );            //实现编码转换
        $obj = new com ( "wscript.shell" );                                   //实例化 com 类
```

```
        $obj->run ( "winrar a -ep " . $objfile . " " . $sourcefile . "", 1, false );          //调用类中的方法，执行文件压缩操作
        echo "<script>alert('文件压缩完成!');</script>";
}
?>
```

秘笈心法

心法领悟 248：将上传文件转换成 ZIP 文件。

将上传文件转换成 ZIP 文件与将上传文件转换成 RAR 文件相同，都是运用 winrar 中 a、-ep 和-df 命令，具体步骤可以参考实例 247。

实例 249　PHP 中解压 ZIP 文件　高级

光盘位置：光盘\MR\04\249　趣味指数：★★★★☆

实例说明

既然可以完成 ZIP 文件的压缩，那么一定也可以实现对 ZIP 文件的解压操作。本实例将介绍如何实现对 ZIP 文件的解压。运行本实例，首先选择要解压的 ZIP 文件，并在文本框中输入要解压文件的存储目录，单击“解压”按钮即可，运行结果如图 4.30 所示。



图 4.30　ZIP 文件解压

关键技术

解压 ZIP 文件同样应用 PHP 的预定义类 com，具体参考实例 247 的关键技术部分。

设计过程

具体设计过程请参见实例 247。

秘笈心法

心法领悟 249：解压 ZIP 文件时，压缩文件和解压文件的存储路径中不可以有空格。

在执行解压 ZIP 文件的操作时，压缩文件和解压文件的存储路径中不可以有空格，即不允许出现 C:\Documents and Settings\或者 E:\Program Files\此类目录，否则解压操作将不会正确执行。

4.6　目 录 操 作

在网站的服务器上，目录是必不可少的，因为它可以把文件进行分类存放，方便管理者查找，所以对目录进行在线管理是非常必要的。

实例 250	目录操作汇总 光盘位置：光盘\MR\04\250	高级 趣味指数：★★★★☆

实例说明

在前面的实例中，已经涉及到很多目录操作的方法，如创建目录、获取当前目录、删除目录等。在本实例中，将对目录的基本操作进行一次汇总，让读者对目录操作方法有一个系统的了解。运行本实例，可以实现目录的创建、浏览和删除操作，其运行结果如图 4.31 所示。



图 4.31　目录操作汇总

关键技术

本实例主要应用 mkdir()、is_dir()、getcwd()、rmdir()，opendir()和 readdir()函数等，完成目录的判断、创建、打开、读取和删除操作。所使用函数的功能和语法如表 4.2 所示。

表 4.2　目录操作函数汇总

名　称	语　法	功　能
mkdir()	bool mkdir(string pathname [, int mode])	新建一个由 pathname 指定的目录。参数 mode 指定目录的模式，在 Windows 下被忽略、自 PHP 4.2.0 起成为可选项。 默认的 mode 是 0777，意味着最大可能的访问权
is_dir()	bool is_dir (string filename)	如果文件名存在并且为目录则返回 TRUE。如果 filename 是一个相对路径，则按照当前工作目录检查其相对路径
getcwd()	string getcwd (void)	返回当前的工作目录
rmdir	bool rmdir (string dirname)	删除 dirname 所指定的目录。 该目录必须是空的，而且要有相应的权限。如果成功则返回 TRUE，失败返回 FALSE

续表

名　称	语　法	功　能
opendir()	resource opendir (string path [, resource context])	打开一个目录句柄，成功返回目录句柄的 resource，失败返回 FALSE。返回值可用于 closedir()、readdir()和 rewinddir()函数中。如果参数path不是一个合法的目录或者因为权限限制或文件系统错误而不能打开目录，opendir() 返回 FALSE 并产生一个 E_WARNING 级别的 PHP 错误信息。可以在 opendir() 前面加上“@”符号来抑制错误信息的输出
readdir()	string readdir (resource dir_handle)	从目录句柄中读取条目，成功则返回目录中下一个文件的文件名，否则返回 FALSE。其参数 dir_handle 是目录句柄的 resource，之前由 opendir() 打开

设计过程

（1）创建 index.php 文件。通过 switch()语句编写一个网页框架，根据超链接中传递的参数值，在 index.php 页面中完成创建目录、浏览目录、删除目录和删除文件功能之间的跳转，其关键代码如下：

```
<?php
switch ($_GET ['dir']) {
 case "create" :
        include ("create.php");
        break;
 case "look" :
        include ("look.php");
        break;
 case "deletedir" :
        include ("deletedir.php");
        break;
 case "deletefile" :
        include ("deletefile.php");
        break;
 default :
        include ("look.php");
        break;
}
 ?>
```

（2）创建 create.php 文件，应用 mkdir()函数完成目录的创建，其关键代码如下：

```
<?php
        if ($_POST ['Submit'] != "") {
                $file_dir = iconv ( "utf-8", "gb2312", $_POST ['objfile'] );
                if (is_dir ( $file_dir )) {
                        echo "<script>alert('目录已经存在!');</script>";
                } else {
                if (mkdir ( $file_dir )) {
                                echo "<script>alert('目录创建成功!');</script>";
                        } else {
                                echo "<script>alert('目录创建失败!');</script>";
                        }
                }
        }
        ?>
```

（3）创建 look.php 文件，通过 getcwd()函数定位到当前目录，并应用 opendir()函数和 readdir()函数读取当前目录中的内容，其代码请参考本书光盘。

（4）创建 deletedir.php 文件，在 look.php 文件的基础上，为输出的目录创建删除超链接，链接到 delete.php 文件，完成目录的删除操作。

（5）创建 deletefile.php 文件，在 look.php 文件的基础上，为输出的文件创建删除超链接，链接到 delete.php 文件，完成文件的删除操作。

（6）创建 delete.php 文件，分别应用 rmdir()函数和 unlink()函数完成目录和文件的删除。

秘笈心法

心法领悟 250：目录的删除。

在应用 rmdir()函数删除指定的目录时，被删除的路径必须指向空的目录，并且该目录的权限必须合乎要求，否则将返回 FALSE。

实例 251　重新定义目录的名称

光盘位置：光盘\MR\04\251

高级

趣味指数：★★★★☆

实例说明

在对网站进行管理和维护的过程中，经常会修改文件夹的名称，这也是目录的一项基本操作。本实例将介绍更新目录名称的方法。运行本实例，单击当前目录中文件夹后的“重命名”超链接，将进入到如图 4.32 所示的页面，在这个页面中完成对指定目录的重命名操作。



图 4.32　执行目录的重命名

关键技术

目录的重命名同样应用的是 rename()函数，有关重命名的技术请参考实例 234。

设计过程

（1）创建 index.php 文件。输出当前目录的文件和目录信息，并且为输出的目录创建“重命名”的超链接，其传递的参数是指定目录的存储位置和当前目录的位置，其关键代码如下：

```
<table width="650" border="0" cellspacing="0" cellpadding="0">
    <tr>
        <td width="50" height="23"> </td>
        <td width="450"><span class="STYLE1">当前目录：<?php echo getcwd ();      ?></span></td>
    </tr>
   <?php
        $lookdir = getcwd ();                                   //获得当前脚本目录
        $i = 0;
        if (is_dir ( $lookdir )) {                               //检测是否是合法目录
            if ($opendir = opendir ( $lookdir )) {               //打开目录
                while ( $li = readdir ( $opendir ) ) {           //读取目录
    ?>
    <tr>
        <td height="23"> </td>
        <td>
```

```
        <?php
                $i ++;
                echo " $i: " . iconv ( "gb2312", "utf-8", $li );
        ?>
</td>
                <td>
<?php
                if ($li == "." || $li == "..") {
                        echo "  ";
                } elseif (filetype ( $li ) == "file") {
                        echo "  ";
                } else {
                        $cata = getcwd () . "\\" . iconv ( "gb2312", "utf-8", $li );
        ?>
        <a href="rename.php?catalog=<?php echo urlencode ( $cata ); ?>&dir_name=<?php echo urlencode ( getcwd () );?>"title="更新目录名称"
target="_blank">重命名 </a>
        <?php }      ?>
</td>
        </tr>
   <?php      }      }      }      ?>
</table>
```

（2）创建 rename.php 文件，根据超链接传递的参数值，创建 form 表单，完成目录的重新命名提交，并且在 rename.php 文件中完成重命名的操作，其关键代码如下：

```
<?php
if ($_POST['Submit'] == "提交") {
        $old_name=iconv("utf-8","gb2312",$_POST['old_name']);                //获取原始目录名称
        $new_name=iconv("utf-8","gb2312",$_POST['new_name']);                //获取新目录名称
        if (file_exists ( $old_name )) {
                if (rename ( $old_name, $new_name )) {                       //把原目录重新命名
                        echo "<script>alert('目录重命名成功！ '); window.location.href='index.php';</script>";
                }else{
                        echo "<script>alert('目录重命名失败！ '); window.location.href='rename.php';</script>";
                }
        } else {
                echo "<script>alert('目录不存在！ '); window.location.href='index.php';</script>";
        }
}
?>
```

秘笈心法

心法领悟 251：超链接中参数值的传递技巧。

在通过超链接的参数向下一个页面中传递值时，最佳的方法就是应用 urlencode()函数对参数值进行编码，其最大的优点是可以避免传递中文字符串或者空格时出现乱码，同时还隐藏了传递的参数值，有一定的保护作用。

实例 252	获取磁盘分区的大小 光盘位置：光盘\MR\04\252	初级 趣味指数：★★☆

实例说明

通过文件系统函数不但可以对目录、文件进行操作，获取目录、文件的相关信息，而且可以获取磁盘分区的大小。运行本实例，将根据文本框提交的目录，获取该目录所在磁盘分区的大小，以及该目录下的所有文件。运行结果如图 4.33 所示。

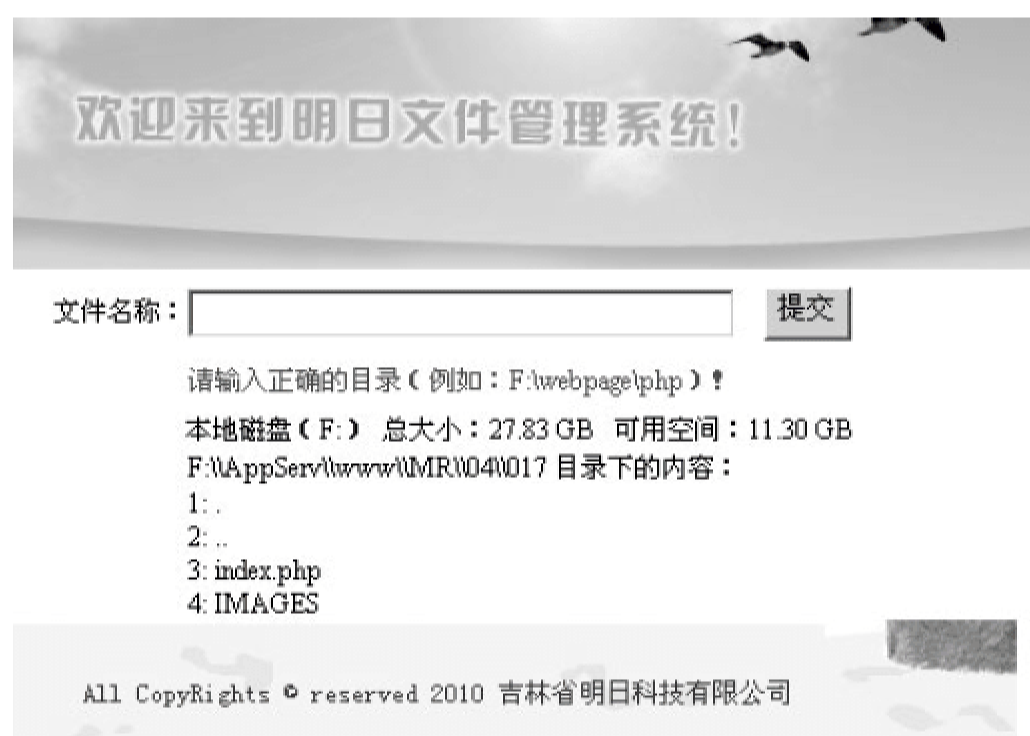


图 4.33　获取磁盘分区的大小

关键技术

获取磁盘分区的大小应用的是 disk_total_space()函数；获取磁盘分区的剩余空间应用的是 disk_free_space()函数。

disk_total_space()函数，获取一个目录的磁盘总大小，语法如下：

```
float disk_total_space ( string directory )
```

该函数根据参数 directory 提供的一个目录字符串，返回相应的文件系统或磁盘分区的所有字节数。

注意：本函数不能作用于远程文件，被检测的文件必须通过服务器的文件系统访问。

disk_free_space()函数，获取一个目录的可用空间，语法如下：

```
float disk_free_space(string directory)
```

参数 directory 用来指定文件系统或磁盘分区。

说明：本函数返回的是目录所在的磁盘分区的总大小，因此以同一个磁盘分区的不同目录作为参数所得到的结果完全相同。

设计过程

（1）创建 index.php 文件。

（2）添加表单，设置文本框、提交按钮，使用 POST 方法，将数据提交到本页。

（3）通过$_POST()获取表单提交的目录路径，首先，判断获取的目录是否合理。然后，通过 iconv()函数对获取的字符串进行编码转换。接着，对获取的字符串进行截取，获取该目录所在的磁盘分区，并应用 disk_total_space()函数获取磁盘分区的大小。最后，应用 opendir()函数和 readdir()函数读取提交目录下的内容。

index.php 的关键代码如下：

```
<?php
if ($_POST ['file_name'] != "" && is_file ( $_POST ['file_name'] ) == false) {
	$file_name = iconv ( "utf-8", "gb2312", $_POST ['file_name'] );                //编码格式转换
	if (file_exists ( $file_name )) {                                              //判断目录是否存在
		$len = strripos ( $file_name, ":" );                                       //截取字符串
		$dir = substr ( $file_name, 0, $len + 1 );                                 //获取提交目录所在磁盘
		$filesize_z=disk_total_space ( $dir );                                     //获取目录总大小
		$filesize_z=number_format($filesize_z/(1024*1024*1024),2,".","");          //数字的格式化
		$filesize_s=disk_free_space ( $dir );                                      //获取磁盘剩余空间
		$filesize_s=number_format($filesize_s/(1024*1024*1024),2,".","");          //字节的格式化
		echo "本地磁盘（" . $dir . "）  总大小：" . $filesize_z . " GB   可用空间：".$filesize_s." GB <br><br>";
		echo $_POST ['file_name'] . " 目录下的内容：" . "<br>";
		$i = 0;
		$list = opendir ( $file_name );                                            //打开目录
		while ( $read_list = readdir ( $list ) ) {                                 //读取目录
```

```
                    $i ++;
                    echo " $i: " . iconv ( "gb2312", "utf-8", $read_list ) . " <br> ";          //输出目录中的内容
                }
                closedir ( $list );                                                              //关闭目录
        } else {
                echo "<script>alert('目录不存在！ ');</script>";
        }
} else {
        echo "<script>alert('请输入正确的目录路径！ ');</script>";
}
?>
```

秘笈心法

心法领悟 252：strripos()函数，判断指定字符串在另一字符串中最后出现的位置。

通过本函数可以获取到指定字符串 A 在另一字符串 B 中最后出现的位置，其返回值为 int 型，根据这个返回值就可以对字符串 B 以字符串 A 为分隔点进行截取，在执行截取的操作时需要应用到 substr()函数。

实例 253　遍历指定目录下的所有文件

光盘位置：光盘\MR\04\253

初级

趣味指数：★★☆

实例说明

在网站的后台管理系统中，经常需要对网站服务器中的文件进行管理和维护，有时需要添加一个文件夹或删除某个文件夹或者文件，为了更有效地查看这些文件或者文件夹，最好的方法就是创建一个文件查询系统，通过它可以查看到指定文件夹下的所有文件。运行本实例，在文本框中输入一个指定的文件夹，单击“提交”按钮，如果该文件夹存在，就可以显示出该文件夹包括的所有文件，运行结果如图 4.34 所示。



图 4.34　遍历指定目录下的所有文件

关键技术

本实例实现的关键是目录和文件处理函数的应用，包括 file_exists()、is_dir()、opendir()、readdir()、和 closedir()函数。

（1）file_exists()函数，检查文件或目录是否存在，语法如下：

```
bool file_exists(string filename)
```

如果由 filename 指定的文件或目录存在则返回 TRUE，否则返回 FALSE。

（2）is_dir()函数，判断给定的文件名是否是一个目录。

```
bool is_dir(string filename)
```

如果文件名存在并且为目录则返回 TRUE。如果 filename 是一个相对路径，则按照当前工作目录检查其相对路径。

注意：本函数的结果会被缓存；本函数不能作用于远程文件，被检查的文件必须通过服务器的文件系统访问。

（3）opendir()函数，打开目录句柄，语法如下：

```
resource opendir ( string path)
```

本函数返回一个目录句柄，可以在 closedir()和 readdir()之前调用。如果 path 不是一个合法的目录或者因为权限限制或文件系统错误而不能打开目录，opendir()返回 FALSE 并产生一个错误信息。

（4）readdir()函数，从目录句柄中读取条目，语法如下：

```
string readdir( resource dir_handle)
```

本函数返回目录中下一个文件的文件名。文件名以在文件系统中的排序返回。

（5）closedir()函数，关闭目录句柄，语法如下：

```
string closedir( resource dir_handle)
```

本函数关闭由 dir_handle 指定的目录流，目录流必须被 opendir()打开过。

设计过程

（1）创建 index.php 文件。添加一个表单，定义表单元素，通过 POST 方法提交目录的路径。

（2）在 index.php 中，判断提交的目录路径是否为空，如果不为空，则打开目录，读取目录中的文件，并输出读取到的文件，关闭目录；否则，输出目录不存在，其关键代码如下：

```
<?php
if(!file_exists ( $look_file )){                                    //检测指定的目录是否存在
        print $look_file."目录不存在!" ;
}else{
        $i = 0;
    if( is_dir ( $look_file )){                                     //检测是否是合法目录
        if ($list = opendir ( $look_file )){                        //打开目录
                while ($read_list = readdir( $list )){              //读取目录
                $i++ ;
                echo " $i: ".iconv("gb2312","utf-8",$read_list)." <br> " ;    //输出目录中的内容
                }
        }
}
    closedir ( $list ) ;                                            //关闭目录
}
?>
```

秘笈心法

心法领悟 253：指定目录下文件的读取、操作。

本实例只是可以浏览到指定目录下包含的文件，不可以对文件进行操作，有关如何操作指定目录下的文件，将在后面的实例中进行讲解。

实例 254　遍历、删除指定目录下的所有文件

光盘位置：光盘\MR\04\254

初级

趣味指数：★★★

实例说明

通过对网站目录的遍历，能够更快地了解网站的结构和网站文件的存储位置；通过对文件内容的遍历，能够掌握网站中每个文件的作用；同时也便于对目录和文件的管理。本实例将介绍如何实现目录和文件的遍历以及删除。运行本实例，如图 4.35 所示。



图 4.35　遍历、操作指定目录下的所有文件

关键技术

遍历、操作指定目录下的所有文件的关键是目录、文件处理函数和字符串处理函数的综合运用。

关键的目录处理函数如下：

（1）chdir()函数，改变当前目录，语法如下：

```
bool chdir(string directory)
```

参数 directory 用于指定更改后目录的位置。操作成功返回 TRUE，否则返回 FALSE。

（2）getcwd()函数，获取当前工作的目录，语法如下：

```
string getcwd( void);
```

（3）rmdir()函数，删除目录，语法如下：

```
bool rmdir ( string dirname )
```

尝试删除参数 dirname 所指定的目录。该目录必须是空的，而且要有相应的权限。如果成功则返回 TRUE，否则返回 FALSE。

在对文件进行操作时，主要应用到如下几个函数：

（1）fopen()函数、fgets()函数、readfile()函数和 fclose()函数，有关这 4 个函数已经在实例 232 和实例 233 中介绍过，这里不再赘述。

（2）filectime()函数，返回指定文件的索引节点修改时间，语法如下：

```
int filectime(string filename);
```

返回文件上次索引节点被修改的时间，如果出错则返回 FALSE。时间以 UNIX 时间戳的方式返回。

（3）filemtime()函数，返回指定文件 filename 的最后修改时间，语法如下：

```
int filemtime(string filename);
```

返回文件上次被修改的时间，失败则返回 FALSE。时间以 UNIX 时间戳的方式返回。

（4）unlink()函数，删除文件，语法如下：

```
bool unlink ( string filename )
```

删除由字符串 filename 指定的文件。成功则返回 TRUE，否则返回 FALSE。

在完成对目录和文件的遍历以及删除过程中，应用到下面几个字符串函数：

（1）substr()函数，从指定的字符串中按照指定的位置截取一定长度的字符，语法如下：

```
string substr(string str,int start,int length)
```

substr()函数的参数说明如表 4.3 所示。

表 4.3　substr()函数的参数说明

参　数	说　明
str	指定字符串对象
start	指定开始截取字符串的位置。如果参数 start 为负数，则从字符串的末尾开始截取
length	可选参数，指定截取字符的个数，如果 length 为负数，则表示取到倒数第 length 个字符

说明：substr()函数中参数 start 的指定位置是从 0 开始计算的，即字符串中的第 1 个字符表示为 0。

（2）strrpos()函数，寻找字符串 haystack 中的字符 needle 最后出现的位置，语法如下：

```
int strrpos(string haystack, char needle)
```

参数 haystack 为寻找的范围；参数 needle 为寻找的字符，其中 needle 只能是一个字符，汉字不可以。如果找不到指定的字符，则返回 FALSE。

（3）strtoupper()函数，将字符串 str 全部转换为大写字符串，语法如下：

```
string strtoupper(string str);
```

（4）htmlentities()函数，将所有 string 的字符都转换成 HTML 的特殊字集字符串，语法如下：

```
string htmlentities(string string);
```

设计过程

（1）创建 index.php 文件。应用目录、文件处理函数和字符串处理函数完成对指定目录下文件的读取操作，并设置查看文件内容的超链接和删除指定文件的超链接。在设置超链接的参数时，应用 urlencode()函数对字符串进行编码，其完整代码请参考本书光盘。

（2）创建 look_file.php 文件，完成对指定文件内容的读取操作，其关键代码如下：

```
<?php
$catalog = iconv ( "utf-8", "gb2312", urldecode ( $_GET [catalog] ) );        //获取文件的信息，设置编码
$filename = iconv ( "utf-8", "gb2312", urldecode ( $_GET [filename] ) );      //获取文件的信息，设置编码
$type = iconv ( "utf-8", "gb2312", urldecode ( $_GET [type] ) );              //获取文件的信息，设置编码
$types = strtoupper ( $type );                                                //完成字符串的大小写转换
if ($types == ".PHP") {                                                       //显示 PHP 文件的内容
 $arr = file ( $catalog . "\\" . $filename );                                 //读取文件
 foreach ( $arr as $value ) {                                                 //循环输出文件内容
        $value = htmlentities ( $value, ENT_COMPAT, "UTF-8" );                //特殊字符的转换
        echo $value . "<br>";                                                 //输出内容
 }
} else {
 $fp = fopen ( $catalog . "\\" . $filename, "r" );                            //显示.txt 文件的内容
 while ( $line = fgets ( $fp ) ) {
        $line = htmlentities ( $line, ENT_COMPAT, "GB2312" );                 //特殊字符的转换
        echo iconv ( "gb2312", "utf-8", $line );                              //输出文件内容
 }
 fclose ( $fp );
}
?>
```

（3）创建 delete.php 文件，完成对指定文件的删除操作，代码请参考本书光盘。

秘笈心法

心法领悟 254：对 URL 字符串进行编码的重要性。

在通过 URL 传递字符串参数时，尽量应用 urlencode()函数对传递的参数值进行编码，这样不但可以保证传递参数值的安全，而且可以防止传递的参数值出现乱码。例如，本实例中在执行指定目录、文件的删除操作时，在执行成功跳转到上级页面时，如果不对 URL 传递的字符串参数值进行编码，那么就不能够完成跳转的操作，会弹出一个错误提示。

第 5 章

会话应用

- COOKIE
- SESSION

5.1　COOKIE

在 Web 技术发展的历史上，COOKIE 技术的出现是一个重大的变革。最先是 Netscape 公司在其 Netscape Navigator 浏览器中引入了 COOKIE 技术，从那时起，World Wide Web 协会就开始支持 COOKIE 标准。以后又经过微软公司的大力推广，即在微软公司的 Internet Explorer 浏览器中完全支持 COOKIE 技术，到现在，绝大多数浏览器都支持 COOKIE 技术，或者至少兼容 COOKIE 技术的使用。

实例 255	控制登录用户的过期时间 光盘位置：光盘\MR\05\255	高级 趣味指数：★★★★

实例说明

COOKIE 是在 HTTP 协议下，将服务器传递给浏览器的少量数据保存到用户浏览器的一种方式。通过这种方式，即使在浏览器被关闭和连接中断的情况下，用户仍然可以维护状态数据。本实例在创建和读取 COOKIE 的同时，设置 COOKIE 的生命周期，实现控制登录用户的过期时间。运行结果如图 5.1 所示。

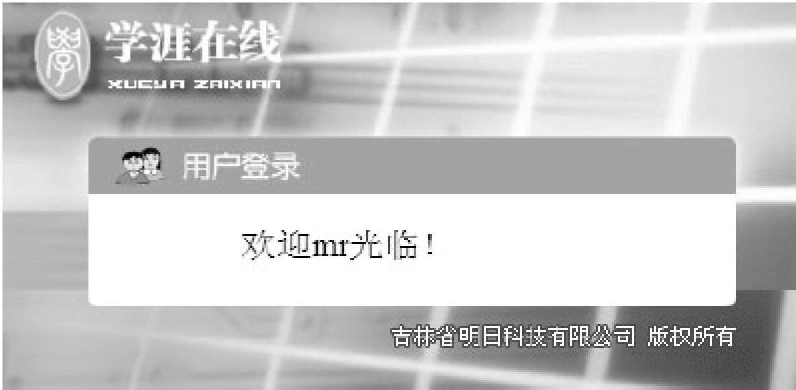


图 5.1　控制登录用户的过期时间

关键技术

（1）setcookie()函数，设置 COOKIE，其代码如下：

```
bool setcookie ( string name [, string value [, int expire [, string path [, string domain [, int secure]]]]]);
```

该函数的参数说明如表 5.1 所示。

表 5.1　setcookie()函数的参数说明

参　数	说　明	举　例
name	COOKIE 的变量名	可以通过 $_COOKIE['cookiename　'] 调用变量名为 cookiename 的 COOKIE
value	COOKIE 变量的值，该值保存在客户端，不能用来保存敏感数据	可以通过$_COOKIE['values ']获取名为 values 的值
expire	COOKIE 的过期时间，expire 是标准的 UNIX 时间标记，可以用 time()函数或 mktime()函数获取，单位为秒	如果不设置 COOKIE 的过期时间，那么 COOKIE 将永远有效，除非手动将其删除
path	COOKIE 在服务器端的有效路径	如果该参数设置为“/”，则它在整个 domain 内有效，如果设置为“/5.1”，它在 domain 下的/5.1 目录及子目录内有效。默认是当前目录
domain	COOKIE 有效的域名	如果要使 COOKIE 在 mrbccd.cn 域名下的所有子域都有效，应该设置为 mrbccd.cn
secure	指明 COOKIE 是否仅通过安全的 HTTPS，值为 0 或 1	如果值为 1，则 COOKIE 只能在 HTTPS 连接上有效；如果值为默认值 0，则在 HTTP 和 HTTPS 连接上均有效

（2）$_COOKIE：经由 HTTP Cookies 方法提交至脚本的变量，用于获取 COOKIE 变量的值。

设计过程

（1）创建 index.php 文件，编写用户登录页面，将用户登录信息提交到 index_ok.php 文件。

（2）创建 index_ok.php 文件，获取表单提交的用户登录信息，并且判断登录的用户名和密码是否正确，如果正确将用户名和密码赋给指定的 COOKIE 变量，并设置 COOKIE 的过期时间，跳转到 cookie.php 页面；否则，直接给出提示信息，并重新跳转到登录页，其关键代码如下：

```
<?php
if($_POST['user']=="mr" && $_POST['pass']=="mrsoft"){                //判断用户名和密码是否正确
 setCookie("user",$_POST['user'],time()+60)or die("禁止 cookie");    //创建 COOKIE
 setCookie("pass",$_POST['pass'],time()+60)or die("禁止 cookie");    //创建 COOKIE
 echo "<script>alert('登录成功！'); window.location.href='cookie.php';</script>";
}else{
 echo "<script>alert('用户名或者密码不正确！'); window.location.href='index.php';</script>";
}
?>
```

（3）创建 cookie.php 文件，判断 COOKIE 变量的值是否存在，如果存在则输出本页内容；否则，给出提示信息并跳转到登录页面，其关键代码如下：

```
<?php
if($_COOKIE['user']=="mr" && $_COOKIE['pass']=="mrsoft"){
        echo "欢迎光临！";
}else{
        echo "<script>alert('COOKIE 已经过期，请重新登录'); window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 255：会话 COOKIE。

如果不设置 COOKIE 的时间，就表示其生命周期为浏览器会话的时间，只要关闭 IE 浏览器，COOKIE 就会自动消失，这种 COOKIE 被称为会话 COOKIE，一般不保存在硬盘上，而是保存在内存中。

实例 256	自动登录 光盘位置：光盘\MR\05\256	高级 趣味指数：★★★★

实例说明

所谓自动登录，即用户第一次成功登录网站后，在一段时间内，再次登录该网站时不再需要填写用户名和密码，而是可以直接进入。

运行本实例，如果是第一次登录，则需要填写用户名和密码才能够登录。如果是再次登录，那么就不需要输入用户名和密码，因为$_COOKIE 会从 COOKIE 中读取这些信息，用户直接单击登录按钮即可。登录成功后进入 main.php 页面，运行结果如图 5.2 所示。



图 5.2　进入网站主页

关键技术

本功能实现的关键是 COOKIE 的创建、过期时间的设置和 COOKIE 值的获取。其中创建 COOKIE 应用的是 setcookie()函数，有关该函数的的应用、过期时间的设置和 COOKIE 值的获取请参考实例 255 中的关键技术，这里不再赘述。

设计过程

（1）创建 index.php 文件，创建表单，添加用户名、密码文本框和保存时间选择单选按钮。最关键的是编写 PHP 脚本，通过$_COOKIE 获取 COOKIE 中存储的用户名和密码数据，并将其作为用户名和密码的默认值，其关键代码如下：

```
<form id="form1" name="form1" method="post" action="index_ok.php">
<input id="lgname" name="name" value="<?php echo $_COOKIE['name'];?>" type="text" class="txt"/>
<input id="lgpwd" name="pwd" value="<?php echo $_COOKIE['pwd'];?>" type="password" class="txt"/>
<input name="times" type="radio" value="3600" checked="checked" />1 小时
<input type="radio" name="times" value="86400" />1 天
<input type="image" name="imageField" src="images/dl.gif" />
<input type="image" name="imageField2" onclick="form.reset();return false;" src="images/cz.gif" />
</form>
```

（2）创建 index_ok.php 文件，通过$_POST[]方法获取表单中提交的数据，验证用户输入的用户名和密码是否正确。如果正确，则通过 setcookie()函数创建 COOKIE，存储用户名和密码，并根据表单提交的时间设置 COOKIE 的过期时间，并跳转到 main.php 页面；如果不正确，则给出提示信息，并跳转到 index.php 页面，其关键代码如下：

```
<?php
if (! empty ( $_POST ['name'] ) and ! empty ( $_POST ['pwd'] )) {                //判断用户名和密码是否为空
 if ($_POST ['name'] == "mr" && $_POST ['pwd'] == "mrsoft") {
        setcookie ( "name", $_POST ['name'], time () + $_POST ['times'] );        //设置 COOKIE 的有效时间为 1 小时
        setcookie ( "pwd", $_POST ['pwd'], time () + $_POST ['times'] );          //设置 COOKIE 的有效时间为 1 小时
        echo "<script>alert('succeed!');window.location.href='main.php';</script>";
 } else {
        echo "<script>alert('false!');window.location.href='index.php';</script>";
 }
} else {
        echo "<script>alert('NULL');window.location.href='index.php';</script>";
}
?>
```

（3）创建 main.php 文件，首先根据$_COOKIE 获取的 COOKIE 值判断用户是否具有访问权限，如果有则可以看到本页内容，否则将给出提示信息，跳转到 index.php 页面，其关键代码如下：

```
<?php
if($_COOKIE[name]==""){        //根据 COOKIE 的值，判断浏览者是否具有访问该页面的权限
 echo "<script>alert('您不具有访问该页面的权限！'); window.location.href='index.php';</script>";
}else{                         //如果正确则输出主页内容
?>
<!--省略了 HTML 内容-->
<?php
}
?>
```

秘笈心法

心法领悟 256：COOKIE 的优点。

在使用 COOKIE 时，如果设置了 COOKIE 的过期时间，浏览器将会保存 COOKIE 数据，即使用户重新启动计算机，只要没有过期，COOKIE 数据就一直有效。

实例 257	单击登录 光盘位置：光盘\MR\05\257	高级 趣味指数：★★★★

实例说明

现在越来越多的企业、个人喜欢用互联网来进行沟通和交流，诸如留言板、聊天室、论坛和博客等交流和展示自己的方式层出不穷，而且一个企业或者个人可能拥有上述多种交流方式。因为每种交流方式都具有自己

的身份验证机制，这样势必造成如果某位用户要以会员的身份访问网站，需要在每个网站上注册，通过身份验证后，才能以会员的身份访问网站。即使用户以同样的用户名与密码在每个网站上注册，这样虽然可以避免忘记或混淆各用户名和密码，但是如果用户在某一时间段内，需访问网站的多个页面或者在不同网站间跳转，仍然需要登录后才能以会员的身份访问网站的话，不仅给用户带来了不便，而且也浪费了服务器的空间。

如果所有的网站能够实现单点登录，不仅在用户体验方面有所提高，而且真正体现了集团多个网站的“兄弟”性。通过这种有机结合，不但便于对网站进行管理，而且能更好地体现公司大平台、大渠道的理念。同时，这样做也利于网站的相互促进与相互宣传。

运行本实例，如果是第一次登录，那么将进入登录页面，登录成功后进入博客主页，如图 5.3 所示；单击博客主页中的“我的论坛”超链接，将以此用户身份直接进入论坛系统主页，如图 5.4 所示；同样，单击论坛系统主页中的“博客”超链接可以跳转到博客主页面。

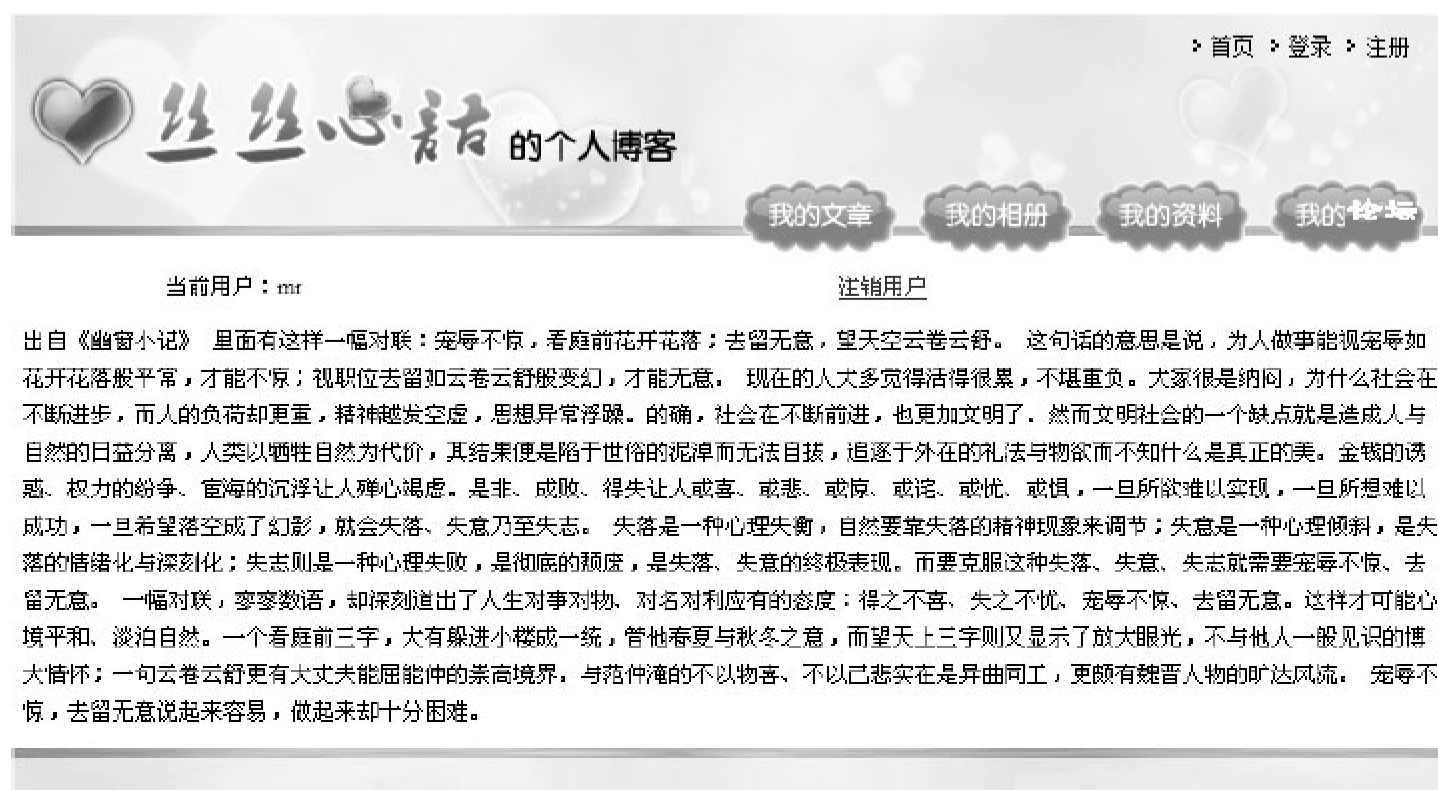


图 5.3　进入博客主页面



图 5.4　进入论坛系统主页

关键技术

本实例开发一个博客网站和一个论坛网站，通过 COOKIE 实现单击登录的功能，即无论在博客或者论坛网站中，只要登录后，此用户名在一定时间内，在这两个网站中是通用的。

其关键就是对 COOKIE 值的判断，如果 COOKIE 的值存在并且与数据库中存储的值相同，那么就可以实现不同系统之间的跳转，否则就必须先登录，才可以访问系统中的内容，其关键代码如下：

```
<?php
include_once 'conn/conn.php';                                //执行连接数据库的操作
if($_COOKIE[name]!="" and $_COOKIE[id]!=""){
        $sql = "select * from tb_login where name = '".$_COOKIE[name]."' and login_id = '".$_COOKIE[id]."'";
        $num = $conne->getRowsNum($sql);                     //返回查询结果
        $conne->close_rst();                                 //关闭数据库
        if($num != 0 or $num != ''){                         //如果在数据表存在该记录，则直接跳转到登录页面
                header("Location: default.php");
        }else{                                               //否则输出登录页面的内容
?>
<!--省略了部分代码-->
<?php
        }
}else{
//清除数据库中 1 小时以前存储的用户登录数据（即 session 过期以后的数据）
$datetimes=mktime()-3600;                                    //获取 1 小时前的时间戳
$sqls="delete from tb_login where datetime<'".$datetimes."'";     //执行删除操作
$nums = $conne->uidRst($sqls);                               //返回查询结果
?>
  <!--省略了部分代码-->
<?php } ?>
```

设计过程

这里以博客网站为例，讲解具体的操作步骤：

（1）创建 index.php 页面，添加用户登录的表单，将登录信息提交到 index_ok.php 页面中。在 index.php 页面中首先连接数据库，然后判断 COOKIE 的值是否为空，如果值不为空，并且在指定的数据表中存在该记录，那么将直接进入网站的主页，不必进行登录操作；如果值为空，则输出网站登录页面，并且删除指定数据表中已经过期的用户登录信息，其代码请参考关键技术中的内容。

（2）创建 index_ok.php 文件，获取表单中提交的登录信息，完成登录操作。单击登录的第一个关键点：将登录成功的用户名和 Session ID 值保存到 COOKIE 中，设置 COOKIE 的过期时间为 1 小时，有效范围是服务器的根目录；单点登录的第二个关键点：将登录的用户名和 Session ID 值存储到一个数据表中。此 COOKIE 值和数据表中的数据是验证当前用户是否具有单点登录权限的唯一依据，代码如下：

```
<?php
session_start();
include_once 'conn/conn.php';                                //执行连接数据库的操作
$name = addslashes($_POST['name']);                          //获取用户名
$pwd = $_POST['pwd'];                                        //获取密码
if(!empty($name) and !empty($pwd)){
        $sql = "select * from tb_member where name = '".$name."' and password = '".$pwd."'";
        $num = $conne->getRowsNum($sql);                     //返回查询结果
        $conne->close_rst();                                 //释放查询结果
        if($num == 0 or $num == ''){                         //如果不正确
                echo "<script>alert('用户名和密码不正确！'); window.location.href='index.php';</script>";
        }else{                                               //如果正确，则将登录用户名数据存储到 COOKIE 中
                $session_id=session_id();                    //获取 SESSION ID
                setcookie("name", $name, time()+3600, "/");  //创建 COOKIE
                setcookie("id", md5($session_id), time()+3600, "/");    //创建 COOKIE
$sqls="insert into tb_login(name,login_id,datetime)values('".$name."','".md5($session_id)."','".mktime()."')"; $nums = $conne->uidRst($sqls);
                                                             //返回查询结果
                echo "<script>alert('登录成功！'); window.location.href='default.php';</script>";
        }
}
?>
```

（3）创建 default.php 网站主页面。其关键就是根据 COOKIE 值和指定数据表中存储的数据判断当前用户是否具有访问主页的权限，如果有则输出网站主页的内容；否则将跳转到登录页面，代码如下：

```
<?php
if($_COOKIE[name]!="" and $_COOKIE[id]!=""){
        include_once 'conn/conn.php';                        //执行连接数据库的操作
        $sql = "select * from tb_login where name = '".$_COOKIE[name]."' and login_id = '".$_COOKIE[id]."'";
```

```
        $num = $conne->getRowsNum($sql);                //返回查询结果
        $conne->close_rst();                            //关闭数据库
        if($num == 0 or $num == ''){
                header("Location: index.php");
        }else{
?>
//省略了主页面代码!
<?php
        }
}else{
        header("Location: index.php");
}
?>
```

（4）在 conn 文件夹下创建 conn.php 文件，实现连接数据库的操作。

秘笈心法

心法领悟 257：单击登录的注意事项。

如果当前用户手动清除了客户端 COOKIE 的信息，那么本实例的功能就会失效，需要重新登录；如果当前用户的客户端禁止了 COOKIE 功能的使用，那么本实例就不能实现单击登录的效果。

实例 258	统计用户在线时间 光盘位置：光盘\MR\05\258	高级 趣味指数：★★★★

实例说明

所谓用户在线时间是指从用户登录到页面开始，到关闭浏览器为止的时间，而 COOKIE 的生命周期在默认情况下关闭浏览器时自动删除，这非常符合在线统计时间的特点，是很多服务类网站优先选择的统计手段。

本实例通过创建 COOKIE，保存用户登录到页面的时间戳，在用户单击退出按钮时计算用户在线时间。运行结果如图 5.5 所示。

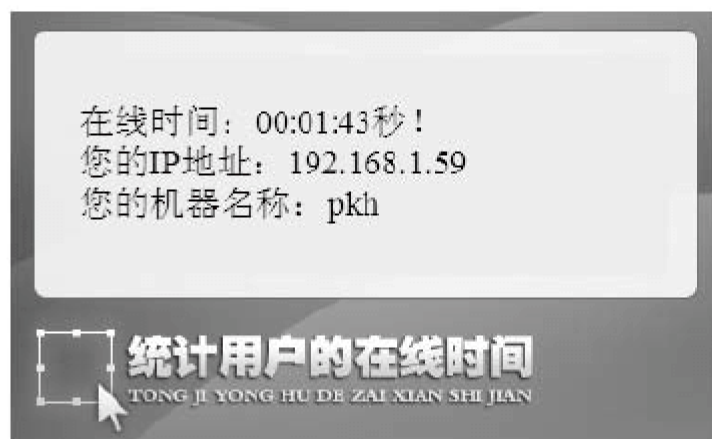


图 5.5　统计用户的在线时间

关键技术

本实例的关键点是通过创建 COOKIE 保存用户登录到页面的时间戳并与结束时间戳做减法运算。

用户登录时间：

```
$str = time();
setcookie('times',$str);
```

用户退出时间：

```
$string = time()-$_COOKIE['times'];
echo "在线时间：".date("H:i:s",$string)."秒！"."<br>";
```

设计过程

（1）沿用实例 257 的 Forum 系统，在 index.php 文件中创建 COOKIE，存储用户的登录时间戳，其关键代

码如下：

```
<?php
include_once 'conn/conn.php';                    //执行连接数据库的操作
$str = time();
setcookie('times',$str);
if($_COOKIE[name]!="" and $_COOKIE[id]!=""){
        $sql = "select * from tb_login where name = '".$_COOKIE[name]."' and login_id = '".$_COOKIE[id]."'";
        $num = $conne->getRowsNum($sql);          //返回查询结果
        $conne->close_rst();                      //关闭数据库
        if($num != 0 or $num != ''){
                header("Location: default.php");
        }else{
?>
```

（2）创建 information.php 文件，应用系统的当前时间戳减去 COOKIE 中存储的登录时间戳，即获取到系统在线时间的时间戳，并通过 date()函数对这个时间戳进行格式化，输出用户的在线时间，同时输出登录用户的 IP 和机器名称，其关键代码如下：

```
<?php
        $string = time()-$_COOKIE['times'];
        echo "在线时间：".date("H:i:s",$string)."秒！"."<br>";
        echo "您的 IP 地址：".getenv('REMOTE_ADDR')."<br>";
        echo "您的机器名称：".gethostbyaddr(getenv('REMOTE_ADDR'));
?>
```

秘笈心法

心法领悟 258：COOKIE 会一直伴随网站中的每个请求，直到过期为止。

浏览器创建一个 COOKIE 后，对于每一个针对该网站的请求，都会在 Header 中带有这个 COOKIE，而且浏览器会一直发送，直到 COOKIE 过期为止。不过，对于其他网站的请求，COOKIE 不会一起发送。

实例 259	限制用户访问网站的时间 光盘位置：光盘\MR\05\259	高级 趣味指数：★★★★

实例说明

默认情况下，COOKIE 的生命周期是以关闭浏览器为基准的，也就是说如果用户不关闭浏览器并且没有对 COOKIE 进行设置，COOKIE 永远也不会过期失效。假如用户在互联网发布的网站有成百上千的用户浏览，并且在线浏览用户数量一直增加，如果不对用户访问网站的时间进行限制，结果只能是服务器资源耗尽，网站瘫痪。

本实例通过设置 COOKIE 限制用户访问网站的时间，运行结果如图 5.6 和图 5.7 所示。



图 5.6　COOKIE 未失效，访问网站



图 5.7　COOKIE 失效

关键技术

本实例的关键点是通过 setcookie()函数创建 COOKIE 并设置 COOKIE 的有效时间。然后，通过判断 COOKIE 变量是否存在和 COOKIE 变量的值是否为空来控制用户访问网站的时间，其关键代码如下：

```
<?php
session_start();                          //初始化 SESSION 变量
$session_id=session_id();                 //获取 Session _ID
setcookie("start",$session_id,time()+10);  //定义 COOKIE 变量，设置 COOKIE 的有效时间是 10 秒
?>
```

设计过程

（1）创建 index.php 文件。首先，初始化 SESSION 变量，获取 SESSION_ID，然后，通过 setcookie()函数创建 COOKIE，并将 SESSION_ID 作为 COOKIE 值，同时设置 COOKIE 的有效时间为 10 秒。

（2）最后，在页面中通过判断 COOKIE 变量的值是否为空来限制用户访问网站的时间，其关键代码如下：

```
<?php
if(isset($_COOKIE['start'])|| $_COOKIE['start']==$session_id){
?>
...//省略了网页中的代码
<?php
}else{
echo "<h1 style ='color:red;'>您访问网站的时间到了</h1>";
}
?>
```

秘笈心法

心法领悟 259：使用 COOKIE 存在隐患。

COOKIE 是以一种客户端明文文件的方式存储的，这样会涉及到一个安全问题，所以在使用 COOKIE 保留用户资料时 CGI 会有一个加密过程。

实例 260	用户来访时客户端创建 COOKIE 光盘位置：光盘\MR\05\260	高级 趣味指数：★★★★

实例说明

COOKIE 是用来解决 HTTP Web 协议无状态的问题的，可以作为一种暂时性的短信息存储器，存储变量值或者属性，其最主要的应用是保存客户信息，例如用户名、密码、联系方式、家庭住址等。

本实例实现用户注册的功能，并通过 COOKIE 获取用户注册的用户名、密码，其运行结果如图 5.8 所示。

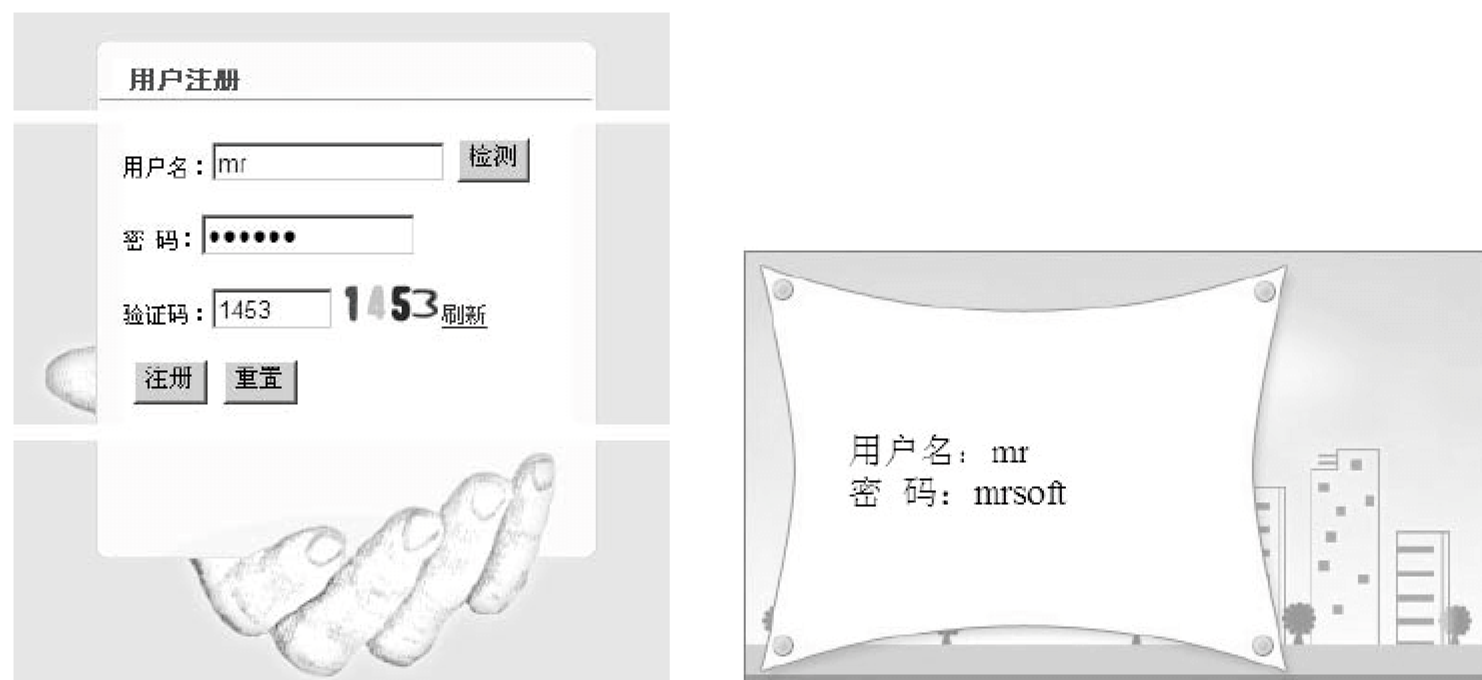


图 5.8　用户注册时客户端获取的 COOKIE 数据

关键技术

本实例实现简单的用户注册功能，其关键是运用 setcookie()函数创建 COOKIE 变量，将用户的注册信息存储到客户端，客户端 COOKIE 的数据存储于如图 5.9 所示的文件中。



图 5.9　客户端生成的 COOKIE 文件

通过 setcookie()函数创建 COOKIE，存储用户名和密码，并设置密码的有效时间。最后，通过自动全局变量$_COOKIE 获取并输出 COOKIE 变量中存储的数据。

设计过程

（1）创建 index.php 文件，添加表单、表单元素，完成用户注册页面的设计。同时能够对提交的用户名进行检测，对验证码进行刷新操作，其关键代码如下：

```
<form name="form1" method="post" action="index_ok.php">
<input name="name" type="text" value="<?php echo $_COOKIE['name'];?>"   size="15">
<input name="check" type="submit" id="check" value="检测" />
        <input name="pwd" type="password" size="15" value="<?php echo $_COOKIE['pwd'];?>" />
<input name="ym" type="text" size="5">
<?php
 $array = array(1 =>"pic/1.jpg",2 =>"pic/2.jpg",3 =>"pic/3.jpg",4 =>"pic/4.jpg",5 =>"pic/5.jpg");
 for($a = 0;$a < 4;$a++){                                    //执行 for 循环，获取数组中的图片数据
        $rand .= rand(1,5)." ";                          //获取随机数据
 }
 $rands = explode(" ",$rand);                            //通过  分割字符串
 $rande = implode($rands);                                    //读取数组中的数据，返回一个字符串
 $_SESSION['ym']=$rande;                                      //将字符串赋给 SESSION 变量
 for($b = 0;$b < 4;$b++){                                    //执行 for 循环
        echo "<img src=pic/".$rands[$b].".jpg>";              //输出验证码图像
 }
 ?>
 <a href='index.php'>刷新</a>
 <br><br>
   <input type="submit" name="sub" value="注册" />
   <input type="reset" name="res" value="重置" />
 </form>
```

（2）创建 index_ok.php 文件，获取表单中提交的数据，完成用户注册的操作。首先，初始化 SESSION 变量，连接数据库。然后，对表单中提交的数据进行判断，判断用户注册信息是否为空及用户名是否被占用。最后，执行 insert 添加语句，将用户注册的信息添加到数据表中，并且通过 setcookie()函数创建 COOKIE 变量存储用户名和密码。index_ok.php 的关键代码如下：

```
if ($_POST ['sub']) {                                          //判断按钮的值，执行不同的操作
 if ($_POST ['name'] == "" || $_POST [pwd] == "" || $_POST [ym] == "") {   //判断注册信息是否为空
        echo "<script>alert('注册信息不能为空');location.href='index.php'</script>";
 }
 $sql = "SELECT * FROM tb_form where name='".$_POST['name']."'";          //定义查询语句
 $rs = mysql_query ( $sql, $conn );                           //执行查询操作
 $result=mysql_num_rows($rs);                                 //获取查询结果
 if($result>0){                                               //如果查询结果大于 0 说明用户名被占用
        echo "<script>alert('用户名被占用');location.href='index.php';</script>";
 }
 if ($_POST ['ym']==$_SESSION['ym']) {                        //判断验证码是否正确
        mysql_query ( "INSERT INTO tb_form(name,pwd,last_date) VALUES('".$_POST ['name']."','".$_POST[pwd]."',now())" );
```

```
        setcookie('name',$_POST ['name']);                    //创建 COOKIE 变量
        setcookie('pwd',$_POST[pwd],time()+60);               //创建 COOKIE，设置过期时间
        echo "<script>alert('注册成功');window.location.href='main.php';</script>";
} else {
        echo "<script>alert('验证码错误');window.location.href='index.php';</script>";
}
}
```

（3）创建 main.php 文件，注册成功后通过$_COOKIE 全局变量获取 COOKIE 中存储的用户名和密码，其关键代码如下：

```
<?php
echo "用户名：".$_COOKIE['name']."<br>";
echo "密  码：". $_COOKIE['pwd'];
?>
```

（4）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 服务器和 db_database05 数据库的连接。

秘笈心法

心法领悟 260：COOKIE 文件命名。

COOKIE 的概念最早是由 Netscape 在 1994 年提出来的，是指保存在浏览器中的小信息包。更确切地说，COOKIE 是保存在用户硬盘上的由字符串组成的小文本文件。

文本文件的命令格式如下：

```
用户名@网站地址[数字].txt
```

实例 261　删除客户端的 COOKIE

光盘位置：光盘\MR\05\261

高级　趣味指数：★★★★

实例说明

COOKIE 具有一定的时效性，当超过其生命周期后即失效。但是，如果 COOKIE 没有超过其生命周期，那么它存储在客户端的数据就有可能被盗取。所以，为保护信息资源，防止不法用户破译，用户可以在关闭浏览器之前，通过程序自动删除存储在客户端的 COOKIE。本实例通过 setcookie()函数实现删除客户端 COOKIE 的功能，其运行结果如图 5.10 所示。



图 5.10　删除客户端的 COOKIE

关键技术

本实例运用 setcookie()函数的第 3 个参数值，完成客户端 COOKIE 的删除操作。因为 setcookie()函数的第 3 个参数是用来设置 COOKIE 的有效时间的，所以如果设置一个小于系统当前时间的有效期，那么 COOKIE 就已经过期，也就完成了客户端 COOKIE 的删除操作，其关键代码如下：

```
setcookie("name","",time()-1);
```

这里设置 COOKIE 的有效期使用的是 time()函数，即系统当前的时间戳，而具体的有效期值是 time()-1。

设计过程

（1）创建 index.php 文件，编写用户登录页面，并在本页获取用户的登录信息，同时通过 setcookie()函数将用户登录信息存储到 COOKIE 中，并且根据表单中提交的值设置它们的有效期，其关键代码如下：

```
<?php
setcookie("username","$_POST[name]",(time()+$_POST[check]));          //创建用户名的 COOKIE 变量
setcookie("pwd","$_POST[pwd]",(time()+$_POST[check]));                //创建密码 COOKIE 变量
setcookie("IP",$_SERVER['REMOTE_ADDR'],(time()+$_POST[check])); //创建用户 IP 地址 COOKIE 变量
?>
```

（2）创建 main.php 文件，设计网站的主页。首先，判断 COOKIE 变量的值是否为空，如果不为空则可以访问本页的内容，否则将给出提示信息，并跳转到登录页面。然后，在本页设置删除客户端 COOKIE 的超链接，当单击“删除 COOKIE”图片链接时，将跳转到本页，在本页中通过 setcookie()函数重新设置 COOKIE 的过期时间为 time()-1，以删除客户端的 COOKIE，其关键代码如下：

```
<?php
if($_COOKIE['username']!="" &&$_COOKIE['pwd']!=""){
 if(isset($_GET[cookie])){
            setcookie("username","",time()-1);
            setcookie("pwd","",time()-1);
            setcookie("IP","",time()-1);
            echo "<script>alert('COOKIE 已删除，确认退出？');location.href='index.php'</script>";
 }
?>
```

秘笈心法

心法领悟 261：COOKIE 的时效性。

在编程领域，程序员更倾向于 COOKIE 的时效性而不是其保存信息的特性。因为 COOKIE 以明文方式保存信息，为了安全起见，程序员更喜欢用服务器端的 SESSION 机制来保存信息。

实例 262 使客户端浏览器支持 COOKIE

光盘位置：光盘\MR\05\262

高级

趣味指数：★★★★

实例说明

有很多用户提出在自己硬盘上保存不明数据会带来安全隐患，所以用户可以关闭 COOKIE 功能，并且从硬盘上删除 COOKIE。以 IE 浏览器为例，可以选择“工具”菜单中的“Internet 选项”命令，在弹出的对话框中直接删除 COOKIE 文件，如图 5.11 所示。

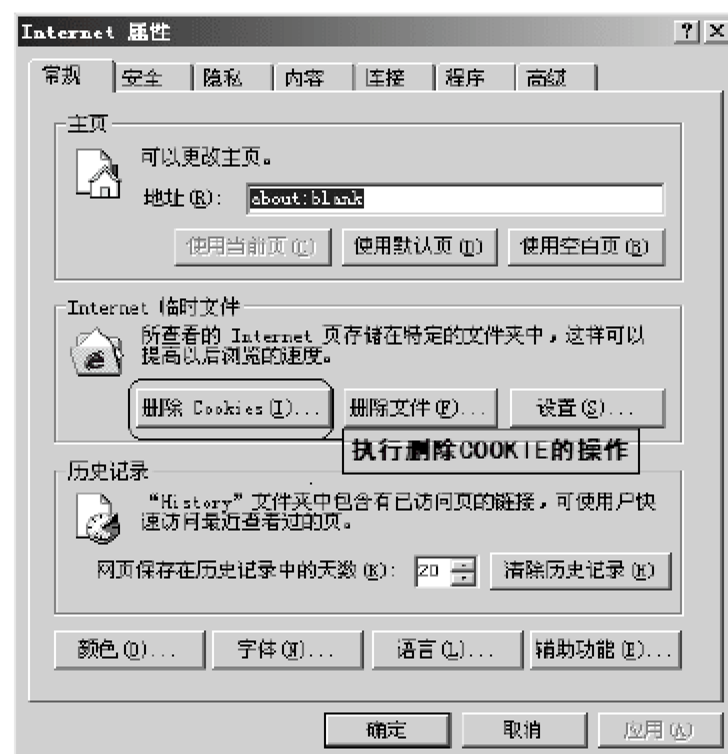


图 5.11　删除 COOKIE

关键技术

本实例的关键点是如何在客户端对 COOKIE 进行控制，其具体的操作方法如下：

以 IE 浏览器为例，选择“工具”菜单中的“Internet 选项”命令，在弹出的对话框中选择“隐私”选项卡，拖动隐私设置级别滑块，根据提示设定想要的隐私级别，如图 5.12 所示：

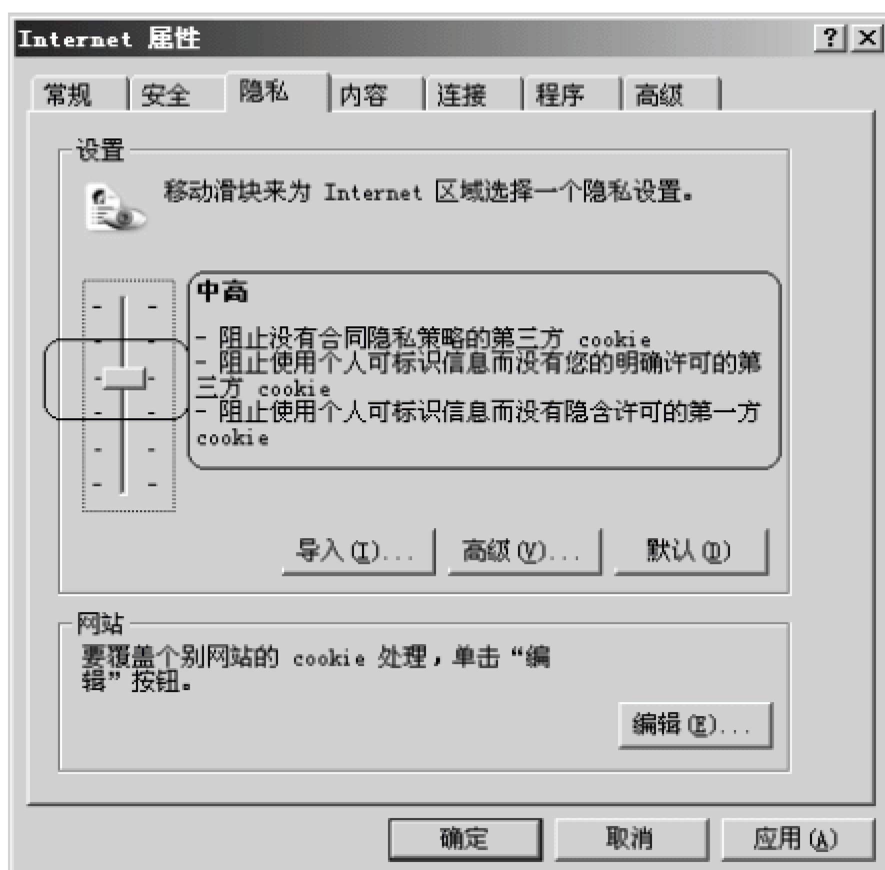


图 5.12　阻止所有 COOKIE

设计过程

这里仍然运用实例 260 的内容，在用户注册成功后通过 COOKIE 变量获取用户的注册信息。在本实例中，调整客户端 COOKIE 的控制权限（方法参见关键技术），然后运行实例 260 的内容，测试在客户端禁止 COOKIE 的情况下，是否可以获取用户注册成功后 COOKIE 变量中存储的用户注册信息。

本实例的实现步骤请读者参考实例 260，其完整代码请参考本书光盘，这里不再赘述。

秘笈心法

心法领悟 262：使用 COOKIE 的注意事项。

（1）在应用 setcookie()函数之前，不能有任何 HTML 输出，如果用户犯了这个错误，那么结果可想而知，随之而来的肯定是一堆错误代码。切记，即使是空格、空行也不可以。

（2）通过 setcookie()函数创建 COOKIE 后，在当前页应用 echo 语句输出$_COOKIE[]时，不会有任何输出。必须在刷新后或者到达下一个页面才可以看到 COOKIE 的值，因为 setcookie()函数执行后，会向客户端发送一个 COOKIE，如果不刷新或者浏览下一个页面，客户端就不能将 COOKIE 送回。

（3）一个浏览器最多能创建 30 个 COOKIE，并且每个 COOKIE 的容量不能超过 4KB；每个 Web 站点能设置的 COOKIE 总数不能超过 20 个。

5.2　SESSION

SESSION 又被称为会话，是指用户访问 Web 站点时的一系列动作。PHP 提供管理会话的机制，可以追踪到用户在这一时期的动作，显示出用户喜欢什么、需要什么等。与 COOKIE 相比，会话更加强大，不但能够管理大量数据，而且可以将信息保存在服务器端，相对更安全，并且没有存储长度的限制。

实例 263 掌控登录用户的权限

光盘位置：光盘\MR\05\263

高级

趣味指数：★★★★

实例说明

在论坛中，如果是管理员，则可以发布公告信息，如果是普通用户，则不可以。那么程序是如何对登录用户的权限进行判断的呢？这就是本实例将要向大家介绍的技术，即通过 SESSION 变量掌控登录用户的权限。如果是管理员登录，则可以添加公告信息，而如果是普通用户，则只可以浏览论坛中的帖子，其运行结果如图 5.13 和图 5.14 所示。



图 5.13　管理员登录



图 5.14　发布留言

如图 5.14 所示，以管理员身份登录到 main.php 主页后，可以单击“留言”超链接，发布留言信息；如果是以普通用户身份登录，那么在 main.php 主页中，“留言”超链接是不可用的，因为普通用户不具备发布留言的权限。

关键技术

（1）应用 SESSION，首先要初始化 SESSION 变量，使用的是 session_start()函数，其语法如下：

```
bool session_start ( void );
```

该函数判断是否有一个会话 ID 存在，如果不存在，就创建一个；如果存在，则将这个已注册的会话变量载入以供使用。

（2）创建 SESSION 变量，应用预定义变量$_SESSION[]，其应用语法如下：

```
$_SESSION["name"] = "value";
```

通过预定义变量$_SESSION[]定义 SESSION 变量。"name"表示设置变量的名称，"value"表示设置变量的值。

设计过程

（1）创建 index.php 文件，添加表单，设计用户登录页面，并获取表单提交的用户名和密码，在本页中对用户提交的数据进行判断。首先，初始化 SESSION 变量，判断用户提交的值是否为空，若不为空，判断用户登录身份，如果是管理员则为 SESSION 变量 type 赋值 0；如果为普通用户则为 SESSION 变量 type 赋值 1，其关键代码如下：

```
<?php
session_start();
if($_POST['sub']==true){
        if($_POST['user']!="" && $_POST['pwd']!=""){                    //判断用户名和密码是否为空
                if($_POST['user']=="mr" && $_POST['pwd']=="mrsoft"){    //判断是否是管理员
                        $_SESSION['type'] = 0;                          //为 SESSION 变量赋值 0
                        echo "<script>alert('管理员登录成功！'); window.location.href='main.php';</script>";
```

```
            }else{
                $_SESSION['type'] = 1;                      //如果不是管理员，为 SESSION 变量赋值 1
                echo "<script>alert('用户名登录成功！'); window.location.href='main.php';</script>";
            }
            $_SESSION['user'] = $_POST['user'];             //将登录用户名存储到 SESSION 变量 user 中
            $_SESSION['pwd'] = $_POST['pwd'];               //将登录密码存储到 SESSION 变量 pwd 中
        }else{
            echo "<script>alert('用户名和密码不能为空！'); window.location.href='index.php';</script>";
        }
}
?>
```

（2）创建 main.php 文件，编写网站的主页面。首先，初始化 SESSION 变量。然后，根据 SESSION 变量 type 的值定义页面的输出内容，如果 SESSION 变量的值为 0，则说明登录用户是管理员，那么具备发布公告的权限；如果 SESSION 变量的值为 1，则说明登录用户是普通用户，那么就不具备发布公告的权限，即控制发布公告的超链接不输出，其关键代码如下：

```
<?php
session_start();                                            //初始化 SESSION 变量
?>
<?php
  if($_SESSION['type']=="0"){                               //根据 SESSION 变量 type 的值决定是否输出发布公告的超链接
?>
<area shape="rect" coords="330,64,385,100" href="mes.php" />
<?php
  }
?>
```

（3）创建 mes.php 文件，编写发布公告信息的页面，根据 SESSION 变量 type 的值决定用户是否具有访问此页面的权限。

秘笈心法

心法领悟 263：SESSION 的生命周期。

SESSION 的生命周期是指从用户访问页面开始到断开与网站的连接为止。在用户每一次连接时，PHP 都会自动生成一个唯一的 SESSION ID 以标识当前用户与其他用户。

实例 264	屏蔽页面刷新对计数器的影响 光盘位置：光盘\MR\05\264	高级 趣味指数：★★★★

实例说明

计数器用来统计一个网站被访问的次数，体现了一个网站受关注的程度。本实例将向大家介绍如何制作计数器，以及如何屏蔽刷新页面对计数器的影响。在本实例中，当用户访问页面时计数器的值会增加一次，但是无论如何刷新页面，计数器的值都不会再次增加，只有重新打开页面才会发生变化，本实例的运行结果如图 5.15 所示。

网站的访问量：73

图 5.15　SESSION 变量屏蔽页面刷新对计数器的影响

关键技术

SESSION 变量控制重复计数的原理如下：在当前页面被访问时，初始化一个 SESSION 变量，判断 SESSION 变量的值是否为空，如果为空，则将计数器的值增加 1，并且将 SESSION 变量赋值为 1。此时，在当前页中 SESSION 变量的值已经不为空，无论如何刷新，其值都不会改变，计数器的值也不会增加。

设计过程

（1）创建 index.php 文件，输出网页当前的访问量。首先，初始化一个 SESSION 变量。然后，判断 SESSION 变量的值是否为空，如果值为空，则打开指定的文本文件，读取其中存储的数据，并且将文本文件中的数据值增加 1，同时将新的数据写入到文本文件中，关闭文件。最后，将 SESSION 变量赋值为 1，其关键代码如下：

```
<?php session_start();
if($_SESSION[temp]==""){                    //判断$_SESSION[temp]==""的值是否为空，其中的 temp 为自定义的变量
 if(($fp=fopen("counter.txt","r"))==false){
        echo "打开文件失败!";
 }else{
        $counter=fgets($fp,1024);           //读取文件中的数据
        fclose($fp);                        //关闭文本文件
        $counter++;                         //计数器增加 1
        $fp=fopen("counter.txt","w");       //以写的方式打开文本文件
        fputs($fp,$counter);                //将新的统计数据增加 1
        fclose($fp);                        //关闭文件
 }
        $_SESSION[temp]=1;                  //登录以后，$_SESSION[temp]的值不为空，给$_SESSION[temp]赋一个值 1
}
?>
```

（2）在 index.php 文件中，创建 img 标签，在 src 属性中调用 gd1.php 文件，完成网站访问量的输出。

（3）创建 gd1.php 文件，通过文件系统函数读取存储在 counter.txt 中的网站访问量的数据，并通过 GD2 函数输出网站访问量的数据。

秘笈心法

心法领悟 264：SESSION 屏蔽页面刷新对计数器影响的弊端。

SESSION 变量防止重复计数的缺点是只对当前页面有效，当重新打开页面时，计数器的值仍然会增加。

实例 265 在不同页面之间传递数据

光盘位置：光盘\MR\05\265

高级

趣味指数：★★★★

实例说明

SESSION 是一种服务器端的会话机制，也就是说在用户与网站断开连接之前，分配给用户的 SESSION ID 不会改变，数据可以在不同页面之间相互传递。这样的机制不仅大大减少程序员的工作量，还使程序运行效率得到提升，使代码结构更加清晰。本实例通过 SESSION 机制实现在不同页面之间传递数据，其运行结果如图 5.16 所示。



图 5.16　在不同页面之间传递数据

关键技术

在本实例中，SESSION 技术的应用如下：

首先，通过 session_start()函数初始化 SESSION 变量。然后，应用 session_set_cookie_params()函数设置 SESSION 的过期时间。接着，应用$_SESSION[]全局变量定义 SESSION 变量的值。最后，应用 session_destroy()函数删除 SESSION 变量。

session_set_cookie_params()函数，设置 SESSION 的过期时间，其语法如下：

```
void session_set_cookie_params( int lifetime [, string path [, string domain [, bool secure]]])
```

session_set_cookie_params ()函数参数的说明如表 5.2 所示。

表 5.2　session_set_cookie_params ()函数的参数说明

参　　数	说　　明
lifetime	必要参数。COOKIE 的生存期
path	可省参数。COOKIE 的有效路径
domain	可省参数。COOKIE 的有效域
secure	可省参数。COOKIE 在安全的范围内被发送

session_destroy()函数，结束当前的会话，其语法如下：

```
session_destroy();
```

该函数清空会话中的所有资源，彻底销毁 SESSION。

设计过程

（1）创建 index.php 文件，设计用户登录页面，并将用户登录信息提交到 index_ok.php 文件进行处理。

（2）创建 index_ok.php 文件，完成对表单中提交数据的处理。首先，应用 session_set_cookie_params()函数根据表单中提交值设置 SESSION 的过期时间，并初始化 SESSION 变量。然后，在本页中获取表单中提交的用户登录信息，判断管理员的用户名和密码是否正确。如果正确，通过$_SESSION[]将用户名和密码存储到指定的 SESSION 变量中，同时跳转到网站主页；如果不正确，则给出提示信息，并跳转到登录页，其关键代码如下：

```
<?php
session_set_cookie_params($_POST['check']);                          //设置 SESSION 的过期时间
session_start();                                                      //初始化 SESSION 变量
if($_POST[sub]){
 if($_POST['user']!="" && $_POST['pwd']!=""){                         //判断用户名和密码是否为空
        if($_POST['user']=="mr" && $_POST['pwd']=="mrsoft"){          //判断管理员用户名和密码是否正确
                $_SESSION['user'] = $_POST['user'];
                $_SESSION['pwd'] = $_POST['pwd'];
                $_SESSION['check'] = $_POST['check'];
                echo "<script>alert('管理员登录成功！');window.location.href='ini.php'</script>";
        }else{
                echo "<script>alert('用户名或者密码不正确！');window.location.href='index.php'</script>";
        }
 }else{
        echo "<script>alert('登录用户名或者密码不能为空！');window.location.href='index.php'</script>";
 }
}
?>
```

（3）创建 ini.php 文件，编写网站的主页面。在网站主页面中应用$_SESSION[]获取 SESSION 变量中存储的用户名、密码和 SESSION 的有效时间，同时创建用户退出的超链接，链接到 logout.php 文件，其关键代码如下：

```
<?php
session_start();
echo "当前时间：". date('Y-m-d H:i:s')."<br>";
echo "用户名：".$_SESSION['user']."<br>";
echo "密码：".$_SESSION['pwd']."<br>";
echo "时效：".$_SESSION['check']."秒<br>";
echo "<a href='logout.php'>退出</a>";
?>
```

（4）创建 logout.php 文件，应用 session_destroy()函数结束当前会话，即退出登录，其关键代码如下：

```
<?php
session_start();
session_destroy();
echo "<script>alert('退出登录！');window.location.href='index.php';</script>";
?>
```

秘笈心法

心法领悟 265：SESSION 的启动。

启动 SESSION 有两种方法：第一种是使用 session_start()函数；第二种是应用 session_register()函数为会话登录一个变量来隐含地启动会话。笔者推荐使用第一种方法，因为第二种方法的运用需要设置 php.ini 文件中 register_globals 的值为 ON，否则使用 session_register()函数启动会话会弹出一个错误信息。

实例 266	SESSION 购物车 光盘位置：光盘\MR\05\266	高级 趣味指数：★★★★

实例说明

购物车是在网上购物时使用的一个临时存储商品的“车辆”，为我们在网上购物提供很大的方便，不用担心一次购买多个商品要进行多次提交结算的操作，可以将商品放入购物车中，等选购完后，一起进行结算。在本实例中，我们将介绍 SESSION 购物车的实现方法，其运行结果如图 5.17 所示。



图 5.17　SESSION 购物车

关键技术

SESSION 购物车，顾名思义，主要是应用 SESSION 变量来实现的。而所谓的购物车就是通过$_SESSION[]创建的两个 SESSION 变量，其中 goodsid 存储商品的 ID， goodsnum 存储商品的数量。

购物车的操作流程如下：首先，登录到网站中浏览商品。然后，购买指定的商品，进入购物车页面，在其中可以实现很多操作，包括更改商品数量、删除商品、清空购物车、继续购物等。最后，填写收货人信息，生成订单，执行订单打印、预览和提交等操作。具体操作流程如图 5.18 所示。

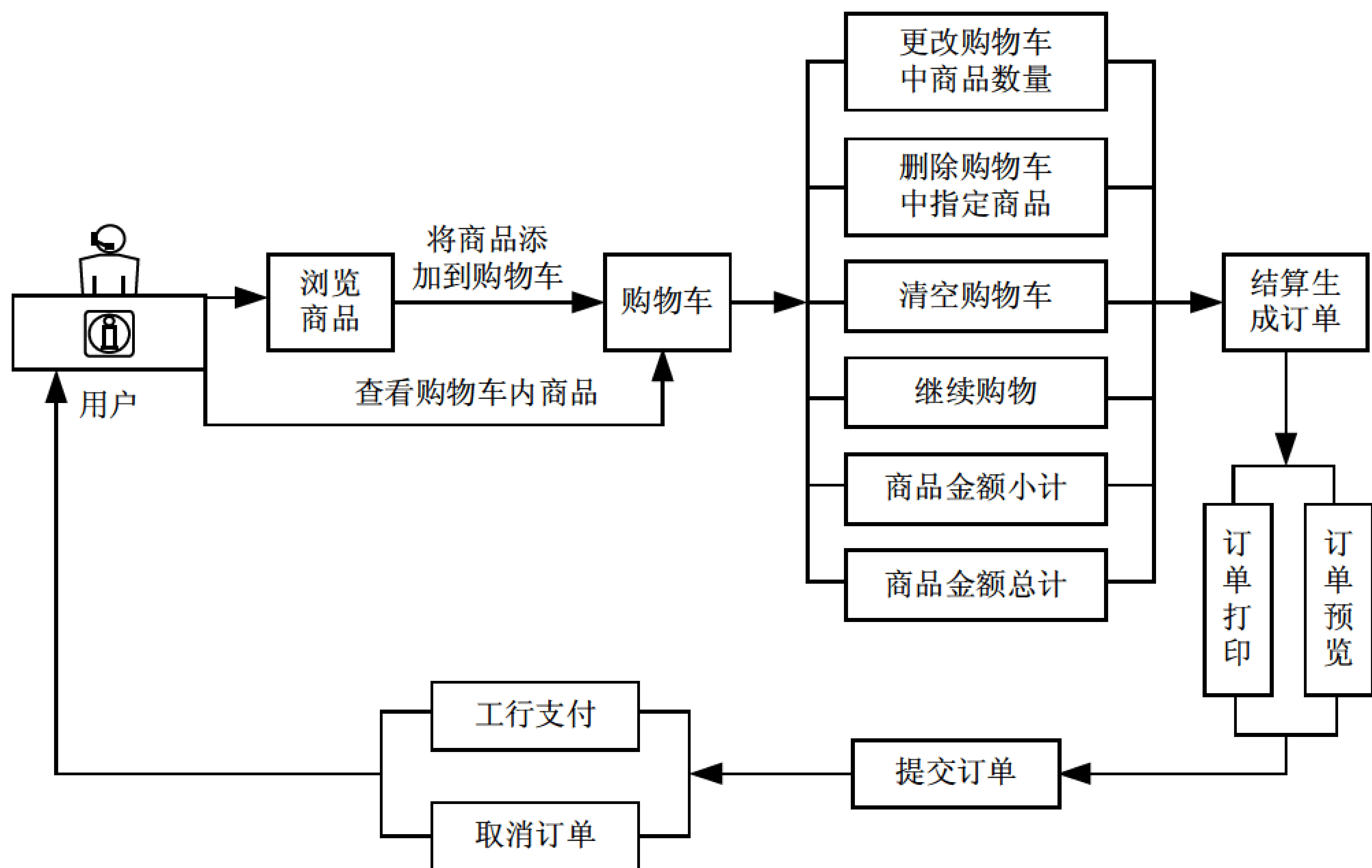


图 5.18　购物车操作流程

设计过程

应用 SESSION 技术开发购物车的步骤如下：

（1）创建 index.php 页面，实现商品展示功能。设计原理是：从数据库中读取商品信息，将商品信息在页面中进行分栏、分页显示，并且为商品设置购买和查看购物车的超链接，运行结果如图 5.19 所示。



图 5.19　购物车商品展示页面

（2）创建 by_commodity.php 文件，实现购买功能，即向购物车中添加商品。首先创建一个购物车，然后判断购物车中是否为空，如果为空，则将商品展示页中“购买”超链接传递的商品 ID（$_GET[id]）和数量添加到购物车中；如果不为空，则判断要添加的商品 ID 是否在购物车中已经存在，如果存在则不能重复添加，否则将跳转到 shopping.php 购物车页面，其代码如下：

```
<?php
session_start();                                                  //初始化 SESSION 变量
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
if($_SESSION["goodsid"]=="" && $_SESSION["goodsnum"]==""){        //判断 SESSION 变量是否为空
      $_SESSION["goodsid"]=$_GET["id"]."@";                       //如果 SESSION 变量为空，则为其赋值为商品的 ID，并以@分隔
 $_SESSION["goodsnum"]="1@";                                      //如果 SESSION 变量为空，则为其赋值为 1，并以@分隔
}else{                                                            //如果 SESSION 变量不为空
        $array=explode("@",$_SESSION["goodsid"]);                 //以@为分隔符，将 SESSION 变量中的数据写入到数组中
  if(in_array($_GET["id"],$array)){                               //判断数组中是否存在指定的 ID
```

```
        echo "<script>alert('该商品已经被放入购物车！');history.back();</script>";
        exit;
    }
     //如果数组中不存在指定的 ID，则说明该商品还没有放入购物车中
     $_SESSION["goodsid"].=$_GET["id"]."@";                          //将该商品添加到购物车中
     $_SESSION["goodsnum"].="1@";                                    //更改商品数量
}
echo "<script>window.location.href='shopping_car.php';</script>";
?>
```

（3）创建 shopping.php 购物车页面，对购买商品进行统计，包括购买商品的数量、每种商品的金额和总金额，并且实现删除购物车中商品、更改购物车中商品数量和清空购物车的功能，同时设置“继续购买”超链接，其代码请参考本书光盘。

（4）创建 change_commodity_counts.php 文件，实现更改购物车中商品数量的功能。首先获取 shopping.php 页面表单中提交的商品数量和商品 ID，然后通过正则表达式验证商品数量是否为正整数，最后根据商品 ID 找到指定的商品，更改商品数量，并返回到购物车页面，其代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=UTF-8" );           //设置文件编码格式
$id=$_POST["id"];                                              //获取商品 ID
$num=$_POST["goodsnum"];                                       //获取商品数量
$preg="/^[0-9]*[0-9]$|^[0-9]*[0-9]$/";                         //编写正则表达式
if($num==""){                                                  //判断提交的值是否为空
 echo "<script>alert('数量不能为空!');history.back();</script>";
 exit;
}else if(!preg_match($preg,$num,$str)){                        //判断提交的数据是否是正整数
        echo "<script>alert('数量只能为正整数!');history.back();</script>";
        exit;
}
$arrayid=explode("@",$_SESSION["goodsid"]);
$arraynum=explode("@",$_SESSION["goodsnum"]);
$key=array_search($id,$arrayid);                               //在数组中搜索给定的值，如果成功则返回相应的键名
$arraynum[$key]=$num;                                          //更改商品数量
$_SESSION["goodsnum"]=implode("@",$arraynum);                  //更改商品数量
echo "<script>window.location.href='shopping_car.php';</script>";
?>
```

（5）创建 delete_commodity.php 文件，实现删除购物车中指定商品的功能。首先获取超链接中传递的商品 ID，然后将 SESSION 中的字符串数据转换成数组，并根据商品 ID 获取指定要删除的商品的键名，最后根据键名为指定的数组赋空值，并重新将数组添加到购物车中，代码请参考本书光盘。

（6）创建 clear_shopping_car.php 文件，实现清空购物车的功能。即将 SESSION 变量$_SESSION["goodsid"]和$_SESSION["goodsnum"]赋空值，其代码如下：

```
<?php
session_start();                                                        //初始化 SESSION 变量
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
$_SESSION["goodsid"]="";                                                //为购物车的 ID 赋空值
$_SESSION["goodsnum"]="";                                               //为购物车数量赋空值
    echo "<script>window.location.href='shopping_car.php';</script>";   //完成清空操作，跳转到购物页面
?>
```

秘笈心法

心法领悟 266：初始化 SESSION 变量和删除 SESSION 变量。

在应用 SESSION 变量实现购物车功能时，最容易出现的一个错误就是忘记初始化 SESSION 变量，如果在文件中没有初始化 SESSION 变量，那么在页面之间就获取不到 SESSION 变量中存储的数据。

在实例 266 中，应用 session_destroy()函数删除 SESSION，而在本实例中通过直接为 SESSION 变量赋空值的方法清空 SESSION，还可以通过 unset()函数删除指定的 SESSION 变量。

```
unset ( $_SESSION['user'] ) ;
```

也可以通过直接为$_SESSION 赋一个空数组的方式清空 SESSION。

```
$_SESSION = array() ;
```

实例 267　解决 SESSION 中的常见问题　高级

光盘位置：光盘\MR\05\267　趣味指数：★★★★

实例说明

SESSION 的功能十分强大，这一点在业界中是公认的。但是越是功能强大或者技术含量高的方法往往执行条件也越苛刻，SESSION 技术的应用也是如此。在本实例中我们将解决一些在 SESSION 应用中常见的问题，包括 session_start()函数的应用限制(其错误使用的运行结果如图 5.20 所示)。和使用 COOKIE 函数设置 SESSION 过期时间时需要注意的事项。



图 5.20　session_start()函数使用限制

关键技术

（1）session_start()函数：使用此函数之前浏览器不能有任何输出，否则会输出如图 5.20 所示的错误信息。

（2）使用 COOKIE 函数为 SESSION 设定过期时间时，首先，要确定浏览器支持 COOKIE，其次，要将 session_start()函数放置在 setcookie()函数之前，否则客户端不会创建 COOKIE 文件。

设计过程

（1）创建 index.php 文件，编写 PHP 脚本，并应用 echo 语句在 session_start()函数之前输出一个字符串，其代码如下：

```
<?php
echo "mrsoft";                                    //输出一个字符串
session_start();                                  //初始化 SESSION 变量
?>
```

当运行本代码时，将输出如图 5.20 所示的错误信息，将 echo 语句放置于 session_start()函数之后即可解决该问题。

（2）利用 COOKIE 机制设置 SESSION 的过期时间。在 index.php 文件中，首先初始化 SESSION 变量，然后定义一个 SESSION 变量 pic，最后通过 setcookie()函数设置 SESSION 变量的过期时间，其代码如下：

```
<?php
    session_start();                              //启动 SESSION
    $_SESSION['pic'] = "mrsoft";                  //设置 SESSION 变量
    setcookie("pi","$_SESSION['pic']",time()+60); //设置 SESSION 的过期时间
?>
```

运行本代码，可以在客户端的 COOKIE 文件夹下查看到创建的 COOKIE 文件。但是，如果将 setcookie()函数放置在 session_start()函数之前，运行本代码，在客户端的 COOKIE 文件夹中找不到对应的 COOKIE 文件。

秘笈心法

心法领悟 267：通过 session_set_cookie_params()函数延长 SESSION 的生命周期。

在开发程序的过程中可能会遇到这样的情况，以会员身份登录一个网站后，若停留的时间较长，再在这个网站中发布信息时，会提示“您没有登录，请登录后再发布信息”，之所以会出现此类问题，是因为在登录时定义的 SESSION 变量已经过期，导致当前用户不可用，需要重新登录。解决此问题最好的方法就是通过 session_set_cookie_params()函数延长 SESSION 的生命周期，即应用 session_set_cookie_params()函数将 SESSION 的过期时间设置为 0。读者不妨自己试一试，看是否能够延长 SESSION 的生命周期。

实例 268	控制页面的访问权限 光盘位置：光盘\MR\05\268	高级 趣味指数：★★★★

实例说明

在 Web 网站的开发过程中，需要对不同的登录用户设置不同的权限。如果是管理员，则可以登录网站后台管理系统，管理网站的数据；如果是普通用户，则只有浏览网站的权限，不可以进入网站的后台管理系统。在本实例中，当输入正确的用户名 mr、密码 mrsoft 时，将输出如图 5.21 所示的页面。

如果直接访问 main.php 页面，即以普通用户的身份访问，那么将输出如图 5.22 所示的提示信息，并跳转到首页。



图 5.21　管理员登录成功展示的页面



图 5.22　普通用户访问 main.php 的效果

关键技术

本实例的关键是在 main.php 文件中判断 SESSION 变量的值是否是管理员的用户名和密码，如果是，则可以访问本页面，否则将给出提示信息，并跳转到登录页面，其关键代码如下：

```
<?php
session_start();                                                    //初始化 SESSION 变量
if($_SESSION['user']=="mr" || $_SESSION['pass']=="mrsoft"){         //判断 SESSION 变量的值是否正确
?>
…//省略了部分代码
<?php
}else{                                                              //如果值不正确，则跳转到首页
	echo "<script>alert('您不具备访问本页面的权限!');window.location.href='index.php';</script>";
}
?>
```

设计过程

（1）创建 index.php 文件，编写用户登录页面，将用户登录信息提交到 index_ok.php 文件中。

（2）创建 index_ok.php 文件，初始化 SESSION 变量，通过$_POST[]方法获取表单提交的用户名和密码，

完成对用户名和密码的验证。如果正确，则将用户名和密码赋给 SESSION 变量，并通过 JavaScript 脚本跳转到 main.php 页面；否则，通过 JavaScript 脚本给出提示信息，跳转到 index.php 页面，其代码如下：

```
<?php
session_start();                                                    //初始化 SESSION 变量
if($_POST['user']=="mr" && $_POST['pass']=="mrsoft"){               //判断提交的用户名和密码是否正确
$_SESSION['user']=$_POST['user'];                                   //如果正确，将其赋给 SESSION 变量
$_SESSION['pass']=$_POST['pass'];
    echo "<script>alert('欢迎您的到来!');window.location.href='main.php';</script>";
}else{
    echo "<script>alert('您输入的用户名和密码不正确!');window.location.href='index.php';</script>";
}
?>
```

（3）创建 main.php 页面，初始化 SESSION 变量，通过$_SESSION[]方法获取 SESSION 变量的值，并判断其是否为真，如果为真，则输出该页面的内容，否则，通过 JavaScript 脚本给出提示信息，跳转到 index.php 页面。其代码可以参考本实例的关键技术。

秘笈心法

心法领悟 268：通过 SESSION 控制网站后台管理系统的权限。

本实例介绍的是通过 SESSION 控制用户访问权限的方法。实例中只是对一个简单的 main.php 页面设置访问权限，在实际的程序开发过程中，可以将其扩展到整个网站的后台管理系统中，即对网站后台管理系统中的所有文件增加权限的访问控制，从而确保后台管理系统不被普通用户访问，保证网站数据的安全。

实例 269　将 SESSION 数据存储到数据库中

光盘位置：光盘\MR\05\269

高级

趣味指数：★★★★

实例说明

将 SESSION 数据变量存储于服务器端是一种较安全的做法，但是设想一下，像校内网这样的日访问量过亿，拥有用户 1500 万的大型网站，如果将所有用户的 SESSION 数据全部存储在服务器端，将消耗巨大的服务器资源。所以程序员在制作大型网站时将 SESSION 存储于服务器端虽然安全，但却不是很好的选择。如果将 SESSION 数据存储于数据库中，那么就可以减轻服务器的压力，同时数据也是比较安全的。这就是本实例将要介绍的内容——如何将 SESSION 数据存储于数据库中，其运行结果如图 5.23 所示。

SESSION

SESSION_ID	SESSION_VALUES	SESSION_TIME
7stli3cuok8p7n61790qb8s727	user|s:2:"mr";pwd|s:6:"mrsoft";	2010-03-30 04:36:12
7stli3cuok8p7n61790qb8s727	user|s:2:"mr";pwd|s:6:"mrsoft";	2010-03-30 04:41:30

图 5.23　将 SESSION 存储到数据库

关键技术

将 SESSION_ID 存储于数据库中，应用的是 session_set_save_handler()函数。该函数调用自定义函数，完成将 SESSION_ID 存储于数据库中的操作，其语法如下：

```
bool session_set_save_handler ( string open, string close, string read, string write, string destroy, string gc)
```

session_set_save_handler()函数的参数说明如表 5.3 所示。

表 5.3　session_set_save_handler()函数的参数说明

参　数	说　明
open(save_path,session_name)	找到 session 存储地址，取出变量名称
close()	不需要参数，关闭数据库
read(key)	读取 session 键值，key 对应 session_id
write(key,data)	其中 data 对应设置的 session 变量
destroy(key)	注销 session 对应的 session 键值
gc(expiry_time)	清除过期的 session 记录

这是一个非常特殊的函数，因为一般函数的参数都是变量，但是此函数的参数为 6 个函数。

说明：在本实例中，编写了 6 个自定义函数作为 session_set_save_handle()函数的参数，自定义函数的语法请参考设计过程。

设计过程

（1）创建 index.php 文件，编写用户登录页面，将数据提交到 index_ok.php 文件中。

（2）创建 index_ok.php 文件，编写 6 个自定义函数作为 session_set_save_handle()函数的参数，同时 session_set_save_handle()函数将表单中提交的用户名和密码赋给 SESSION 变量，登录成功后跳转到 main.php 页面，其关键代码如下：

```
<?php
function _open(){
	global $conn;
	$conn = mysql_connect ( "localhost", "root", "111" ) or die ( "连接 mysql 出现错误" );
	mysql_select_db ( "db_database05", $conn ) or die ( "连接数据库出现错误" );
	mysql_query ( "SET NAMES utf8" );				//设置数据库编码
	return(true);
}
function _close(){						//关闭数据库
	global $conn;
	mysql_close($conn);
	return(true);
}
function  read($key){
	global $conn;						//全局变量$conn 连接数据库
	$time = time();						//设定当前时间
	$sql = "select session_data from tb_session where session_key = '$key' and session_time > $time";
	$result = mysql_query($sql,$conn);
	$row = mysql_fetch_array($result);
	if ($row){
		return($row['session_data']);			//返回 SESSION 名称及内容
	}else{
		return(false);
	}
}
function _write($key,$data){	//首先判断 SESSION 是否失效，如果失效则重新向数据库中添加数据
	global $conn;
	$time = 60*60;						//设置失效时间
	$overdue_time = time() + $time;				//得到 UNIX 时间戳
	$sql = "select session_data from tb_session where session_key = '$key' and session_time > $overdue_time";
	$result = mysql_query($sql,$conn);
	if (mysql_num_rows($result) == 0 ){			//没有结果
		$sql = "insert into tb_session values('$key','$data',$overdue_time)";	//插入数据库 sql 语句
		$result = mysql_query($sql,$conn);
	}else{							//否则，更新 SESSION 的数据
		$sql = "update tb_session set session_key = '$key',session_data = '$data',session_time = $overdue_time where session_key = '$key'";
								//定义更新语句
```

```
				$result = mysql_query($sql,$conn);							//执行更新语句
		}
		return($result);											//返回标识
}
function _destroy($key){
		global $conn;
		$sql = "delete from tb_session where session_key = '$key'";					//删除数据库 sql 语句
		$result = mysql_query($sql,$conn);
		return($result);
}
function _overdue($expiry_time){
		global $conn;
		$overdue_time = time();										//将参数$expiry_time 赋值为当前时间戳
		$sql = "delete from tb_session where $expiry_time < $overdue_time";			//删除数据库 sql 语句
		$result = mysql_query($sql,$conn);
		return($result);
}
session_set_save_handler('_open','_close','_read','_write','_destroy','_overdue');
session_start();
if($_POST['sub']=="sub"){
if($_POST['user'] == "" || $_POST['pwd'] == ""){
			echo "<script>alert('用户名或者密码不能为空！ '); window.location.href='index.php';</script>";
}else{
			$_SESSION['user'] = $_POST['user'];
			$_SESSION['pwd'] = $_POST['pwd'];
			echo "<script>alert('登录成功！ '); window.location.href='main.php';</script>";
}
}
?>
```

（3）创建 main.php 文件，连接数据库，通过 while()语句循环输出数据库中存储的 SESSION 数据。

（4）创建 conn 文件夹，编写 conn.php 文件，连接 MySQL 数据库服务器和 db_database05 数据库。

秘笈心法

心法领悟 269：注意 SESSION 函数运用时的放置顺序。

在运行 SESSION 函数时要特别注意它们的使用顺序，尤其是 session_start()函数，在运用这个函数之前浏览器不能有任何输出。还有 session_save_path()和 session_set_cookie_params()函数，都必须在 session_start()函数之前使用，否则就没有任何意义。解决这个问题最好的方法就是通过 session_set_cookie_params()函数延长 SESSION 的生命周期。

实例 270	SESSION 更换聊天室界面 光盘位置：光盘\MR\05\270	高级 趣味指数：★★★★

实例说明

SESSION 可以实现数据在页面之间的传递，并且在 SESSION 的生命周期中一直有效。在本实例中，将运用 SESSION 的这个特性，编写一个简单的聊天室换肤功能。在聊天室中，根据提交的颜色值更换聊天室的背景颜色，其运行结果如图 5.24 所示。

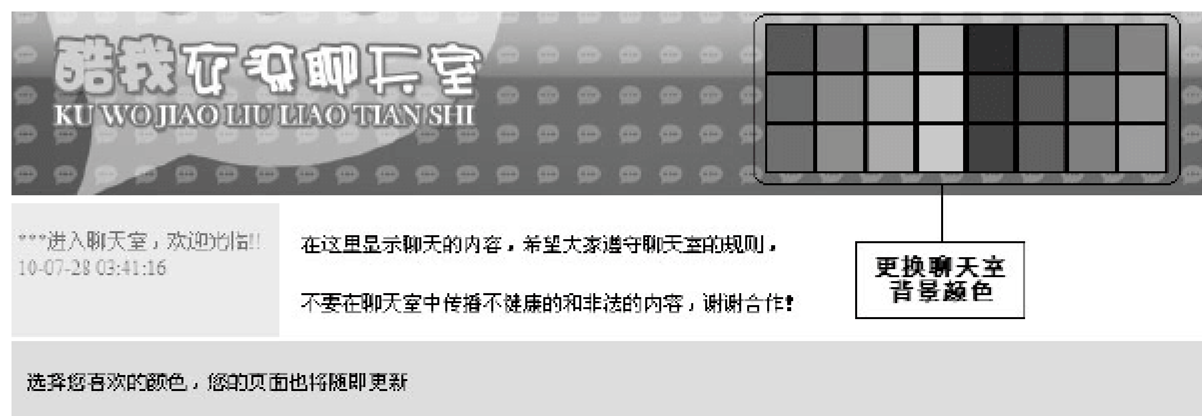


图 5.24　更换聊天室背景颜色

关键技术

本实例中，首先获取超链接中传递的参数值。然后，将参数值赋给指定的 SESSION 变量。最后，将 SESSION 变量设置为页面背景颜色 bgcolor 的值，进而实现聊天室背景颜色的更换，其关键代码如下：

```
<?php
$_SESSION['bgcolor']=$_GET['col'];                    //将超链接传递的参数值赋给 SESSION 变量
?>
<body bgcolor="<?php if($_SESSION['bgcolor']=="){echo "white";}else{echo $_SESSION['bgcolor'];}?>">
```

在 body 标记中，通过 if 语句判断 SESSION 变量的值，如果值为空，则设置 bgcolor 的值为 white；否则，直接将 SESSION 变量作为 bgcolor 的值。

设计过程

（1）创建 index.php 文件，设计一个简单的聊天室页面，在聊天室的 log 中插入一个颜色选项卡图片，并为每种颜色设置矩形热点链接，链接到 index.php 文件，同时定义超链接参数 col，参数值是热点链接对应的颜色值。

（2）在 index.php 文件中，获取超链接传递的参数值，并将其赋给 SESSION 变量，然后根据 SESSION 变量的值设置页面中 body 标记的 bgcolor 的值，完成聊天室背景颜色的更换。index.php 的关键代码如下：

```
<?php
$_SESSION['bgcolor']=$_GET['col'];                    //获取超链接传递的参数值，并赋给 SESSION 变量
?>
<body bgcolor="<?php if($_SESSION['bgcolor']=="){echo "white";}else{echo $_SESSION['bgcolor'];}?>">
//省略了部分代码
<map name="Map" id="Map">
<area shape="rect" coords="4,3,27,26" href="index.php?col=0066FF" />
<area shape="rect" coords="64,5,87,26" href="index.php?col=00CCFF" />
<area shape="rect" coords="31,34,55,56" href="index.php?col=999900" />
<area shape="rect" coords="218,64,237,85" href="index.php?col=CC9933" />
<area shape="rect" coords="184,33,209,55" href="index.php?col=CC6600" />
<area shape="rect" coords="214,7,237,28" href="index.php?col=3399FF" />
<area shape="rect" coords="8,63,27,87" href="index.php?col=996633" />
<area shape="rect" coords="65,61,88,87" href="index.php?col=99CC33" />
<area shape="rect" coords="152,61,177,87" href="index.php?col=CC3333" />
</map>
</body>
```

秘笈心法

心法领悟 270：编程思想比编程技术更重要。

在本实例中只是将聊天室作为实例开发的一个大背景，并没有实现聊天室的功能，其目的是展示如何通过 SESSION 设置聊天室的背景颜色。本方法虽然很简单，但是体现的是页面换肤的基本原理，在具体的实战开发中，读者可以将这个技术进行升华，不但可以改变背景的颜色，而且可以对图片进行更换。程序设计的原理非常简单，难的是如何在原理的基础上让内容更加丰富，这就要求程序开发者有自己独到的开发思想。

实例 271 清理 SESSION 缓存提高网站访问的效率

高级

光盘位置：光盘\MR\05\271

趣味指数：★★★★

实例说明

SESSION 缓存将网页中的内容临时存储到客户端 IE 的 Temporary Internet Files 文件夹下。当网页被第一次浏览后，页面的部分内容在规定的时间内就被临时存储在客户端的临时文件夹中，这样下次访问此页面时，就可以直接读取缓存中的内容，从而提高网站的浏览效率。但是如果不对 SESSION 缓存做定期处理，也会给服务

器带来压力。本实例将讲解 SESSION 缓存的运用和清理方法，其运行结果如图 5.25 所示。



图 5.25　输出缓存信息

关键技术

在本实例中应用 session_cache_limiter()函数创建缓存，应用 session_save_path()函数设置缓存文件的存储路径，应用 session_cache_expiry()函数设置缓存时间。

session_cache_limiter()函数，用于创建 SESSION 缓存，其语法如下：

```
string session_cache_limiter ( [string cache_limiter])
```

参数 cache_limiter 用于设置缓存的方式，参数值说明如表 5.4 所示。

表 5.4　cache_limiter 参数值的说明

参　数　值	说　　明
nochache	不设置缓存
private	私有方式
private nochache	私有方式，但不过期
public	公有方式

session_cache_expire ()函数，用于设置缓存时间，其语法如下：

```
int session_cache_expire ( [int new_cache_expire])
```

参数 cache_expire 为可选参数，设置 SESSION 的过期时间，单位为分钟。默认过期时间是 180 分钟（即不设置参数值的情况下）。

session_save_path()函数，取得或者重新配置目前 SESSION 的存储路径，其语法如下：

```
string session_save_path([string path])
```

如果设置参数 path，表示重新设置 SESSION 的存储路径；如果不设置参数 path，表示直接获取当前 SESSION 的存储路径。

设计过程

（1）创建 index.php 文件。首先，定义 SESSION 临时缓存文件夹路径并开启缓存，设置缓存时间为 30 分钟，启用 SESSION，并设置 SESSION 变量，其代码如下：

```
<?php
$path =   './tmp/';                              //定义缓存文件的临时存储路径
session_save_path($path);                        //设置缓存存储路径
session_cache_limiter('private');                //设置缓存方式
$session_cache = session_cache_limiter();        //开启缓存
session_cache_expire(30);                        //定义缓存时间
$session_expire = session_cache_expire();        //设置缓存时间
session_start();                                 //初始化 SESSION 变量
$_SESSION['cache'] = $session_cache;             //为 SESSION 变量赋值
$_SESSION['expire'] = $session_expire;
?>
```

（2）在 index.php 文件中，获取并输出缓存中存储的数据，创建超链接，执行清理缓存的操作，清理方法是为$_SESSION 赋一个空数组值，并应用 session_destroy()函数彻底删除会话，其关键代码如下：

```
<?php
echo "<a href = 'index.php?cache=0'>清理缓存</a>";
if($_GET[cache] == "0"){
        $_SESSION = array();                                    //清空 SESSION
        session_destroy();                                      //销毁会话
        echo "<script>alert('清理 SESSION 缓存成功');location.href='http://www.mrbccd.com'</script>";
}
?>
```

秘笈心法

心法领悟 271：运用 SESSION 缓存的注意事项。

（1）SESSION 缓存被存储于客户端。如果用户未设定 SESSION 缓存的临时目录，默认情况下存储于客户端 IE 的 Temporary Internet Files 文件夹下。以 XP 系统为例，Temporary Internet Files 文件夹所在路径为 C:\Documents and Settings\Administrator\Local Settings\Temporary Internet Files。

（2）虽然可以通过缓存时间函数设置缓存过期时间，但是在过期之后并不能够删除存储在客户端缓存文件夹中的缓存文件，缓存文件的删除由 Windows 操作系统自行控制，或者由用户手动删除。

（3）session_save_path()、session_cache_limiter()、session_cache_expiry()函数都必须在 session_start()函数之前调用，否则会出现错误。

第 6 章

图形图像处理

- ⏭ 图像与统计
- ⏭ 控制、显示图片特性
- ⏭ Jpgraph 类库
- ⏭ GD2 函数库

6.1　图像与统计

图像与数字相结合，既可以美化页面，同时又达到了突出显示的效果，可谓一举两得。在本节中将向读者展示图像与统计结合的妙用，希望对读者的开发和学习有所帮助。

实例 272	图形计数器 光盘位置：光盘\MR\06\272	中级 趣味指数：★★★

实例说明

网站计数器的形式有很多，图形计数器是比较常用的一种。PHP 制作图形计数器的方法主要有两种，一种是利用 GD2 函数直接绘制图形，另一种是事先将图形用图形绘制工具绘制出来，然后用 PHP 代码进行调用。本例将采用第二种方式实现，统计网页的访问量，其运行结果如图 6.1 所示。

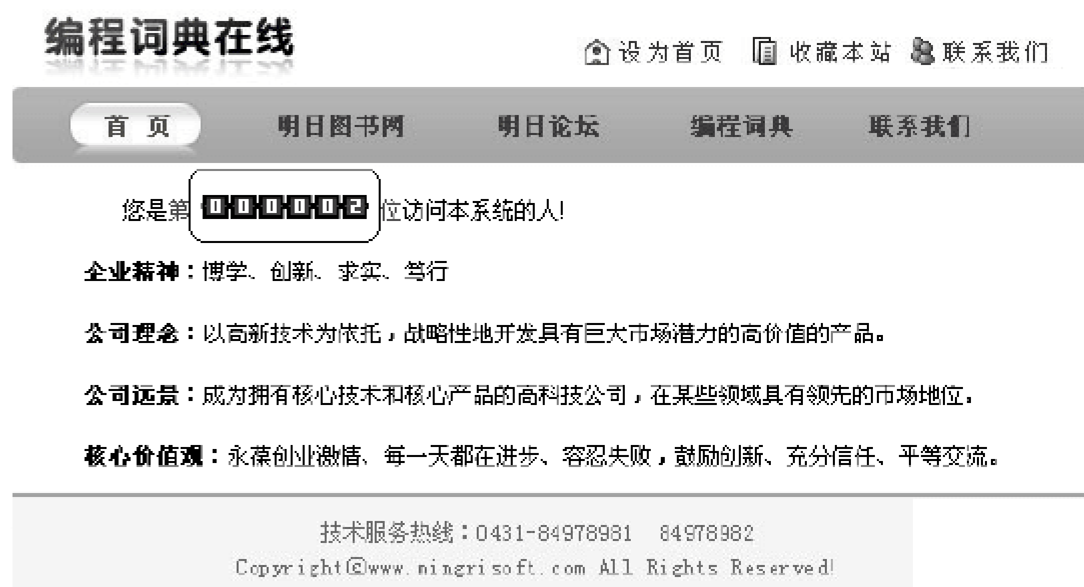


图 6.1　图形计数器

关键技术

图形计数器的设计原理：

（1）为了避免用户通过刷新页面来提高网站的访问量，通过 SESSION 变量防止重复计数。首先，通过判断$_SESSION [temp]的值是否为空决定计数器的值是否加 1，如果该值不为空，说明用户正在访问该网站，这时计数器的值不加 1，反之加 1。

（2）将文本文件中存储的访问量数据以数字图片的形式输出。首先，应用 strlen()函数获取文本文件中数据的长度。然后，定义网站访问量的最大值为 6 位数字（当然位数可以根据实际情况设定），用 0 填充剩余位数。最后，通过 for 循环和 switch 语句，将从文本文件中读取的数据以图形的形式输出，其关键代码如下：

```
$len = strlen ( $counter );                        //获取字符串的长度
$str = str_repeat ( "0", 6 - $len );               //获取 6-$len 个数字 0
for($i = 0; $i < strlen ( $str ); $i ++) {         //获取变量$str 的字符串长度
        $result = $str [$i];
        $result = '<img src=images/0.gif>';
        echo $result;                              //循环输出$result 的结果
}
for($i = 0; $i < strlen ( $counter ); $i ++) {     //获取字符串的长度
        $result = $counter [$i];
        switch ($result) {
                //如果值为"0",则输出 0.gif 图片
                case "0" :
                        $ret [$i] = "0.gif";
                        break;
```

```
            case "1" :
                    $ret [$i] = "1.gif";
                    break;
            case "2" :
                    $ret [$i] = "2.gif";
                    break;
        …//省略部分代码
        }
        echo "<img src=images/" . $ret [$i] . ".>";             //输出文本文件中存储的数据
}
```

设计过程

（1）创建 index.php 页，设计网站的页面，输出公司简介的内容。

（2）在 index.php 页中，通过文件系统函数编写一个文本计数器，将数据存储于文本文件 counter.txt 中，并通过 SESSION 变量屏蔽页面刷新对计数器的影响，其关键代码如下：

```
<?php
session_start ();
if ($_SESSION [temp] == "") {                        //判断$_SESSION[temp]==""的值是否为空，其中 temp 为自定义的变量
 if (($fp = fopen ( "counter.txt", "r" )) == false) {
        echo "打开文件失败!";
 } else {
        $counter = fgets ( $fp, 1024 );              //读取文件中数据
        fclose ( $fp );                              //关闭文本文件
        $counter ++;                                 //计数器增加 1
        $fp = fopen ( "counter.txt", "w" );          //以写的方式打开文本文件
        fputs ( $fp, $counter );                     //将新的统计数据增加 1
        fclose ( $fp );                              //关闭文件
 }
 $_SESSION [temp] = 1;                               //为$_SESSION[temp]赋一个值
}
?>
```

（3）在 index.php 页中，通过文件系统函数读取文本文件中存储的网站访问量的数据，并将数据以图形的形式输出。代码请参考关键技术中的内容。

秘笈心法

心法领悟 272：另一种将数字转换成图形的方法。

首先，应用函数 intval()将计数器的值转变成字符串。设置计数器最多显示 8 位数字（当然位数可以根据实际情况设定），所以根据 strlen()函数确定当前计数值的位数，之后用 8 减去该位数值得出剩余的位数，为了使页面美观，剩余的位数值用 0 填充，最后输出文本文件中统计的数据，其关键代码如下：

```
$imagenum=intval($num);                                    //类型转换
for($i=0;$i<8-strlen($imagenum);$i++){
    echo "<img src=images/count/0.gif>";                   //用 0 填充剩余的位数
}
for($i=0;$i<strlen($imagenum);$i++){                       //通过循环调用图片显示结果
    echo "<img src=images/count/".substr($imagenum,$i,1).".gif>";
}
```

实例 273　GD2 图形计数器

光盘位置：光盘\MR\06\273

高级　趣味指数：★★★★☆

实例说明

本实例向读者介绍另一种计数器，即将统计数据以 GD2 函数生成的图像进行输出，本实例的运行结果如图 6.2 所示。



图 6.2　GD2 图形计数器

关键技术

本实例联合应用文件系统函数和 GD2 函数，完成网站访问量的存储、读取和输出，其原理如图 6.3 所示。

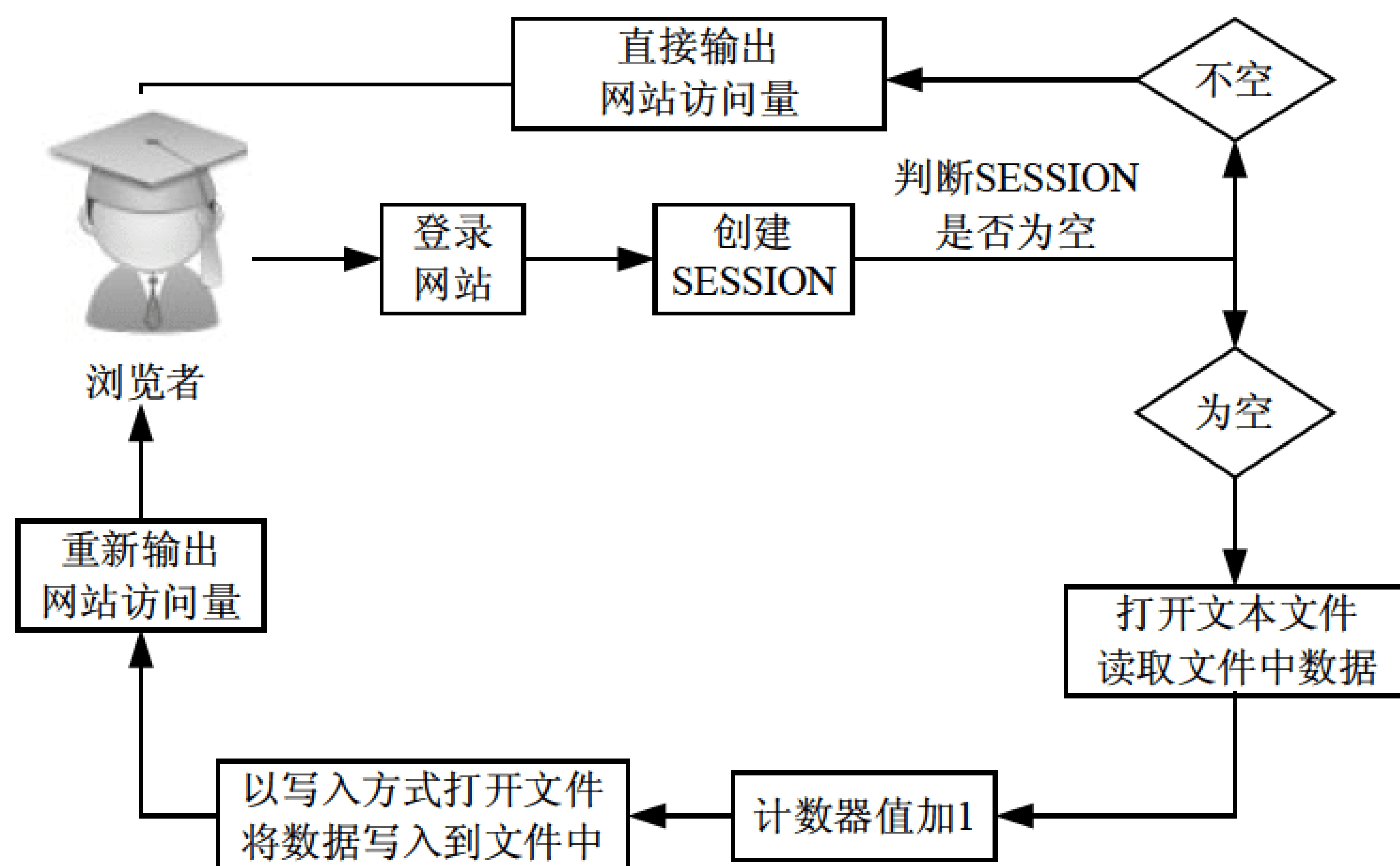


图 6.3　GD2 图形计数器设计原理

在这个通过文本文件统计网站访问量的程序中，概括起来应用了三方面的技术：

第一方面，文件系统操作函数，实现数据的写入、读取功能。

第二方面，SESSION，通过 SESSION 变量防止重复计数。

第三方面，GD2 函数，将访问量数据生成图像，改变数据的输出方式。

设计过程

（1）创建 index.php 文件，完成页面的设计，并通过文件系统函数将网站的访问量添加到文本文件 counter.txt 中，并且通过 SESSION 变量屏蔽刷新功能对计数器的影响。创建 img 标签，链接到 gd2.php 文件，在该文件中将网站访问量的数据生成图像。

（2）创建 gd2.php 文件，应用 GD2 函数将从文本文件中读取到的数据生成图像。其中，关键是运用 imagettftext()和 imagestring()函数，完成对中文字符串和英文字符串的输出，其关键代码如下：

```
<?php
//以图形的形式输出数据库中的记录数
header("Content-type: text/html; charset=UTF-8");                    //设置文件编码格式
if(($fp=fopen("counter.txt","r"))==false){
        echo "打开文件失败!";
}else{
        $counter=fgets($fp,1024);
        fclose($fp);
        $im=imagecreate(240,24);                                      //通过 GD2 函数创建画布
        $gray=imagecolorallocate($im,255,255,255);
        $color =imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255));    //定义字体颜色
```

```
        $text="网站的访问量:";                                  //定义中文字符串
        $font = "Fonts/FZHCJW.TTF";
        imagettftext($im,14,0,20,18,$color,$font,$text);        //输出中文
        imagestring($im,5,160,5,$counter,$color);               //输出网站的访问次数
        imagepng($im);
        imagedestroy($im);
}
?>
```

秘笈心法

心法领悟 273：通过 imagettftext()输出中文字符串。

在通过 imagettftext()函数输出中文字符串时，如果页面使用的是 utf-8 编码，那么可以直接使用这个函数；如果页面中使用的是 gb2312 或者其他编码，那么就必须应用 iconv()函数对要输出的中文字符串的编码进行转换，因为 imagettftext()函数只支持 utf-8 编码。

实例 274	通过图像显示投票统计结果 光盘位置：光盘\MR\06\274	高级 趣味指数：★★★★☆

实例说明

在线调查中的投票系统相信大家都了解，在本实例中，将编写一个简易的投票系统，并将投票结果以图像的形式进行输出，运行效果如图 6.4 所示。

我最喜爱的运动员	选项	投票结果
○	欧阳一	2/2
⊙	南宫一	3/2
	确认提交	

图 6.4　输出投票结果

关键技术

投票系统包括动态生成投票主题、动态添加投票选项内容、投票主题管理和投票功能实现。这里重点介绍在投票功能中如何以图像的形式输出投票结果。

在 look_vote.php 文件中，根据总的投票数，计算出每一项所占的比例，然后根据这个比例计算输出的图像，其关键代码如下：

```
<?php
$sql="select * from tb_vote where tb_vote_id='$_GET[id]'";      //显示指定投票项
$result=mysql_query($sql,$conn);                                //执行查询指定的投票选项
$row=mysql_fetch_array($result);                                //获取查询结果
$s=$row[tb_vote_num];                                           //计算总的投票数量
echo "$row[tb_vote_name]\n";                                    //输出投票选项的名称
?>
<?php
 if($rows[tb_record_num]==0)                                    //判断如果投票数为 0
 $width=0;                                                      //定义变量值为 0
 else
 $width=$rows[tb_record_num]/$s;                                //否则，根据投票数和总的投票数计算输出所占比例
 if($width!=0){                                                 //以图像的形式输出投票所占比例
        echo "<img src=".$rows[tb_record_color].".bmp width=".($width*20)." height=10>\n";
 }
 echo $rows[tb_record_num]."/".$row[tb_vote_num];               //以数字输出投票数和总的投票量
?>
```

■ 设计过程

本实例的文件架构如图 6.5 所示。

conn —— 连接数据库文件夹
css —— CSS样式文件夹
images —— 背景图像文件夹
1.bmp
2.bmp
3.bmp
4.bmp
5.bmp
delete_vote.php —— 删除投票主题
index.php —— 浏览投票主题
login.php —— 管理员登录
look_vote.php —— 提交投票和浏览投票结果
look_vote_ok.php —— 处理提交的投票
vote.php —— 添加投票选项

图 6.5　投票系统的文件架构

说明：本实例编写一个简易的投票系统，具体内容请参考本书光盘，这里只对其中投票结果的图像输出进行讲解。

这里重点介绍 look_vote.php 文件，在该文件中完成投票的提交操作和投票结果的输出。首先，根据超链接传递的 ID，查询出指定的投票主题，并计算出投票总数。然后，创建表单，提交投票选项，将数据提交到 look_vote_ok.php 页进行处理。最后，执行查询语句，从数据库中查询出投票结果，并完成投票结果的输出，其关键代码如下：

```
<?php
include_once ("conn/conn.php");
if (! $_GET [id]) {
 echo "没有指定 ID！ ";
 exit ();
} else {
 ?>
//执行查询，输出投票主题
<?php
 $sql = "select * from tb_vote where tb_vote_id='$_GET[id]'"; //显示指定投票项
 $result = mysql_query ( $sql, $conn );
 $row = mysql_fetch_array ( $result );
 $s = $row [tb_vote_num];
 echo "$row[tb_vote_name]\n";
 ?>
//创建表单，提交投票选项
<form action=look_vote_ok.php method=post>
<input type=hidden name=id value="<?php    echo $row [tb_vote_id]; ?>">
<input type=hidden  name=v_type value="<?php echo $row [tb_vote_type]; ?>">
</form>
//输出投票结果
 <?php
 $sql2 = "select * from tb_vote_record where tb_vote_id='$_GET[id]'";
 $result2 = mysql_query ( $sql2, $conn );
 while ( $rows = mysql_fetch_array ( $result2 ) ) {
        ?>
<tr>
                              <td height="25" align="center">
<?php

        if ($row [tb_vote_type] == 0) {
            echo "<input type=radio name=r value=" . $rows [tb_record_id] . ">\n";
        } else {
            echo "<input type=checkbox name=r[] value=" . $rows [tb_record_id] . ">\n";
        }
        ?></td>
                              <td> <?php    echo $rows [tb_record_name]; ?></td>
                              <td> 
<?php
```

```
        if ($rows [tb_record_num] == 0)
            $width = 0;
        else
            $width = $rows [tb_record_num] / $s;
        if ($width != 0) {
            echo "<img src=" . $rows [tb_record_color] . ".bmp width=" . ($width * 20) . " height=10>\n";
        }
        echo $rows [tb_record_num] . "/" . $row [tb_vote_num];
        ?></td>
                        </tr>
<?php
}
?>
<?php
}
?>
```

秘笈心法

心法领悟 274：如何屏蔽同一个用户的重复投票。

在本实例的简易投票系统中，应用 SESSION 变量来屏蔽同一用户的重复投票，它可以防止用户在一次访问中进行重复投票。不过本系统也存在一个问题，即如果同一用户想再次投票，只要关闭当前页面，重新登录即可。

实例 275　通过图像显示密码安全强度

光盘位置：光盘\MR\06\0275

中级

趣味指数：★★★★☆

实例说明

在本实例中将介绍图像的另一项妙用，即显示注册填写密码的安全强度。运行本实例，当在密码文本框中输入一个密码时，光标离开后，将自动判断这个密码的安全强度，并以图像的形式返回一个判断结果，其运行结果如图 6.6 所示。

图 6.6　通过图像显示密码安全强度

关键技术

对密码安全强度进行判断是在 JavaScript 脚本中，应用正则表达式来完成的，其原理如下：

（1）对密码中的值进行判断，如果只包含数字、英文字符串或者特殊字符串中的任意一种，那么说明密码安全强度差。

（2）对密码中的值进行判断，如果包含数字、英文字符串或者特殊字符串中的任意两种，那么说明密码安全强度中。

（3）对密码中的值进行判断，如果同时包含数字、英文字符串和特殊字符串，那么说明密码安全强度强。

在 JavaScript 脚本中，通过 search()方法调用正则表达式，完成对密码值的判断，其关键代码如下：

```
var p1 = (form.pwd.value.search(/[a-zA-Z]/)!=-1)?1:0;                 //判断密码中是否包含英文字符串
var p2 = (form.pwd.value.search(/[0-9]/)!=-1)?1:0;                    //判断密码中是否包含数字
var p3 = (form.pwd.value.search(/[^a-zA-Z0-9_]/)!=-1)?1:0;            //判断密码中是否包含特殊字符
var p = p1+p2+p3;                                                     //定义判断的返回值
if(p==1){
	chknew_pwd.innerHTML="<img src='images/1_03.jpg'>";               //输出密码安全强度差的图像
}else if(p==2){
	chknew_pwd.innerHTML="<img src='images/2_03.jpg'>";               //输出密码安全强度中的图像
}else if(p==3){
	chknew_pwd.innerHTML="<img src='images/3_03.jpg'>";               //输出密码安全强度强的图像
}
```

设计过程

本实例完成一个用户注册的功能，并且在表单中通过 JavaScript 脚本对提交的元素进行判断，最终符合要求后，将用户注册信息添加到指定的数据表中。

（1）创建 index.php 页面，完成用户注册页面和表单的设计。通过 JavaScript 对表单中提交的值进行判断，并直接将判断的结果反馈到 div 标签中，最终将表单中的数据提交到 index_ok.php 文件中，完成用户注册的操作。

```
<form name="form_reg" method="post" action="index_ok.php" onSubmit="return chkreginfo(form_reg,'all')">
<tr>
	<td height="30"><div align="right">密码：</div></td>
	<td height="30" colspan="2" align="left"> 
<input type="password" name="pwd" id="pwd" size="30" class="inputcss"
				onBlur="chkreginfo(form_reg,1)"> <font color="#FF0000">*</font> 
		<div id="chknew_pwd" style="color: #FF0000"></div>
	</td>
</tr>
<tr>
	<td height="30"> </td>
	<td width="150" height="30"><input type="image" src="images/form (2).jpg">   </td>
	<td width="343"><img src="images/form.jpg" width="72" height="26"
				onClick="form_reg.reset()" style="cursor: hand" /></td>
</tr>
</form>
```

（2）编写 JavaScript 脚本，创建 check.js 脚本文件，完成对表单中提交的值进行判断。

（3）创建 index_ok.php 文件，获取表单中提交的数据，将数据添加到指定的数据表中。

（4）创建 conn.php 文件，完成 MySQL 服务器和 MySQL 数据库的连接操作，并设置编码格式为 utf-8。

秘笈心法

心法领悟 275：正则表达式的运用。

在本实例中，不只一次使用了正则表达式，分别为通过正则表达式对密码的强度进行判断，对手机号码进行判断，对座机号码进行判断以及对邮箱地址进行判断。

```
/^13(\d{9})$|^15(\d{9})$|^189(\d{8})$/                    //判断手机号码
/^(\d{3}-)(\d{8})$|^(\d{4}-)(\d{7})$|^(\d{4}-)(\d{8})$/   //判断座机号码
/\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*/            //判断邮箱
```

实例 276　数字图像验证码

光盘位置：光盘\MR\06\276

高级

趣味指数：★★★★☆

实例说明

为了防止用户通过恶意程序登录站点，提高网站的安全性，项目开发时，在用户登录、注册以及发表主题等模块中加入验证码模块是必不可少的。应用验证码技术，可以有效地防止用户通过探测的方式非法登录，从而有效地提高站点的安全性。在本实例中，应用 JavaScript 技术及 0~9 这 10 个别致数字图片为博客的后台登录设计验证码功能，其运行效果如图 6.7 所示。



图 6.7　数字图像验证码

关键技术

首先，通过 JavaScript 中 Math 对象的 random()方法随机生成 4 位随机数，并通过 doument 对象的 write()方法将这 4 位随机数所对应的图片以链接形式输出，链接地址为某 PHP 页面，然后，通过$_GET[]全局数组提取链接地址所传入的随机数值，通过 PHP 文件操作函数读取该随机数对应的图片，并以二进制的形式输出到浏览器。接着，定义一个重新指定图片链接地址的方法，当单击“看不清”链接时，调用该方法生成新的验证码图片。最后，在表单中添加一个隐藏域，用于提交 4 位随机数，当用户单击表单中的“登录”按钮时，通过 JavaScript 将隐藏域的值与验证码文本框所输入的内容进行比较，完成验证操作。

本实例的关键技术主要是通过 JavaScript 随机生成 4 位随机数，以及如何将随机数对应的图片以二进制的形式输出到浏览器。

（1）创建 4 位随机数

通过 JavaScript 中 Math 对象的 random()方法随机生成 4 位随机数，关键代码如下：

```
var num1=Math.round(Math.random()*10000000);
var num=num1.toString().substr(0,4);        //生成 4 位随机数
```

（2）将图片以二进制形式输出

应用 PHP 的文件操作函数即可实现将图片以二进制的形式输出到浏览器，本实例主要应用 fread()函数实现，该函数的语法如下：

```
fread ( int handle, int length)
```

参数说明：

handle：fopen()函数打开文件后所返回的文件句柄。

length：所读取文件的长度 。

例如，本实例中应用 fread()方法将图片以二进制形式输出到浏览器，实现代码如下：

```
$fp=fopen($address,"r");                         //打开文件
echo fread($fp,filesize($address));              //读取文件
```

设计过程

（1）编写 index.php 用户登录页面，创建表单，在表单中添加用户名文本域、密码域、验证码文本域以及保存随机数的隐藏域。

（2）为了提高网站的安全性和人性化，在用户登录或注册表单中的数据提交给服务器之前，需要对其进行合理性验证。本实例应用 JS 文件夹下的 fun.js 脚本文件在客户端对表单提交数据进行验证，这样可以降低服务器负载，在实际项目开发中，对一些安全性要求较高的模块，建议采用客户端和服务端双重验证，这样既可以为用户操作带来方便，还可以保证系统的安全。

（3）编写 JavaScript 脚本生成 4 位随机数，并且在用户单击“看不清”链接时，设法在不刷新页面的情况下动态改变这 4 位随机数的值。本实例通过 Math 对象的 random()方法产生 4 位随机数，然后通过自定义方法 code_1()动态更改验证码的值。实现该过程的关键代码如下：

```
<script language="javascript">
var num1=Math.round(Math.random()*10000000);
 var num=num1.toString().substr(0,4);            //产生 4 位随机数
 document.write("<img name=codeimg4 src='xym.php?num="+num.substr(0,1)+"'>");
 document.write("<img name=codeimg5 src='xym.php?num="+num.substr(1,1)+"'>");
 document.write("<img name=codeimg6 src='xym.php?num="+num.substr(2,1)+"'>");
 document.write("<img name=codeimg7 src='xym.php?num="+num.substr(3,1)+"'>");
 form_login.xym1.value=num;
 function code_1(){                              //通过该方法动态改变验证码的值
 var num1=Math.round(Math.random()*10000000);
        var num=num1.toString().substr(0,4);
        document.codeimg4.src="xym.php?num="+num.substr(0,1);
        document.codeimg5.src="xym.php?num="+num.substr(1,1);
        document.codeimg6.src="xym.php?num="+num.substr(2,1);
        document.codeimg7.src="xym.php?num="+num.substr(3,1);
        form_login.xym1.value=num;               //将验证码的值赋给隐藏域
 }
</script>
```

（4）创建 xym.php 文件，采用 PHP 文件操作函数将图片以二进制的形式输出到浏览器，这样可以有效地提高静态页面与动态页面之间的交互性，其代码如下：

```
$num=$_GET[num];                                 //接收 GET 方法传递过来的参数
 $address="images/code/".$num.".gif";            //获取图片路径
 $fp=fopen($address,"r");                        //打开图片文件
 echo fread($fp,filesize($address));             //将图片文件读取到浏览器
 fclose($fp);                                    //关闭打开的文件句柄
```

秘笈心法

心法领悟 276：PHP 标记的特性。

如果在 PHP 页面中，不仅包含 PHP 代码，而且包含其他语言代码，如 JavaScript、Css 或 Html 等，需要书写完整的 PHP 代码块标记，即“<?php”与“?>”都要写，而若页面代码完全由 PHP 构成，则建议只写开始标记，不写结束标记，这样可以防止恶意用户通过非法注入等手段在页面代码后注入恶意代码。

实例 277	中文图像验证码 光盘位置：光盘\MR\06\277	高级 趣味指数：★★★★☆

实例说明

验证码不但可以使用数字图像来完成，而且可以使用中文图像来完成。为了防止用户通过恶意程序登录站点，提高网站的安全性，在本实例中将介绍如何通过中文图像来生成验证码，运行结果如图 6.8 所示。



图 6.8　中文图像验证码

关键技术

开发中文图像验证码，首先应该确定所使用的文字，并且将文字制作成图片，以数字进行命名。然后，将所使用的文字定义到数组中。最后，计算出数组中元素的长度值，并应用 rand()函数从这个长度值中随机取值，根据随机取值输出对应的中文图像，即验证码，其关键代码如下：

```
<?php
$str = array ("大", "更", "创", "天", "科", "客", "博", "技", "立", "新" );          //定义中文字符串的数组
$word = count ( $str );                                                          //计算出数组中元素个数
for($i = 0; $i < 4; $i ++) {                                                     //执行 for 循环，以数组中元素个数为范围随机取出 4 个值
 $num = rand ( 0, $word - 1 ); //$word=$word*2-1                                //随机取值
//输出随机取出的 4 个值对应的中文图片
 $img = $img . "<img src=' images/checkcode/" . $num . ".gif' width='16' height='16'>";   //显示随机图片
 $pic = $pic . $str [$num];                                                      //将图片转换成数组中的文字
}
?>
```

设计过程

（1）编写 index.php 用户登录页面，创建表单，在表单中添加用户名文本域、密码域、验证码文本域以及保存随机数的隐藏域。

（2）在 index.php 页面中，编写 PHP 脚本，定义验证码所使用的中文字符串，并生成验证码，其关键代码请参考本实例的关键技术。

（3）本实例中沿用实例 276 中对表单中元素进行验证的方法，fun.js 脚本文件同样存储于 js 文件夹中。

（4）创建 checkuser.php 文件，完成对用户提交的登录信息的验证。

秘笈心法

心法领悟 277：如何计算数组中元素的数目。

在本实例中，应用 count ()函数计算数组中元素的数目。注意，count()函数只适用于计算数组中元素的个数，如果将其应用到其他类型的数据中，那么返回值只有一个单元。如果要计算字符串的长度，那么要使用 strlen()函数，切记 strlen()函数不可对数组元素进行操作，虽然不返回错误信息，但是返回的结果是不准确的。

实例 278	缩略图艺术库 光盘位置：光盘\MR\06\278	高级 趣味指数：★★★★☆

实例说明

上传图片是一个比较常见的功能，但是在输出上传图片时可能会遇到一些问题，由于上传图片的大小没有

固定，从而导致图片在输出时变形，这是一个很让人伤脑筋的问题。如果在上传图片时直接将其生成一个固定大小的缩略图，并同时保存上传的原始图片，那么在输出时就不会有任何问题。

在本实例中就实现了这样一个功能，即在将图片上传到服务器的同时，生成图片的缩略图，浏览的是图片的缩略图，而如果要下载，则下载的仍是原始图片。本实例的运行效果如图 6.9 所示。



图 6.9　缩略图艺术库

关键技术

在本实例中，有关多图片的任意上传和图片的分页输出不再是讲解的重点，如果读者要了解这方面的内容，请参考本书第 4 章的内容。

本实例重点讲解的是如何生成图片的缩略图，并且为缩略图添加水印，其原理如下：

（1）通过 GetImageSize()函数获取上传图片的信息，包括大小、类型等，并根据其类型，新建一个对应类型的图像。

getimagesize()函数，获取指定图像的大小，语法如下：

```
array getimagesize ( string filename [, array &imageinfo] )
```

getimagesize()函数将测定任何 GIF、JPG、PNG、SWF、SWC、PSD、TIFF、BMP、IFF、JP2、JPX、JB2、JPC、XBM 或者 WBMP 图像文件的大小，并返回图像的尺寸以及文件类型和一个可以用于普通 HTML 文件中 <IMG> 标记中的 height/width 文本字符串，其返回值是一个具有 4 个单元的数组：

☑ 索引 0 包含图像宽度的像素值；
☑ 索引 1 包含图像高度的像素值；
☑ 索引 2 是图像类型的标记：1 = GIF，2 = JPG，3 = PNG，4 = SWF，5 = PSD，6 = BMP，7 = TIFF(intel byte order)，8 = TIFF(motorola byte order)，9 = JPC，10 = JP2，11 = JPX，12 = JB2，13 = SWC，14 = IFF，15 = WBMP，16 = XBM。
☑ 索引 3 是文本字符串，内容为 height="yyy" width="xxx"，可直接用于 IMG 标记。

如果参数 filename 指定的图像不存在或者其不是有效的图像，将返回 FALSE 并产生一条 E_WARNING 级的错误。

（2）对上传图片的大小与指定缩略图的大小进行比较，看其是否需要创建缩略图。

（3）根据判断结果应用 imagecopyresampled()函数重新生成新的图像。

imagecopyresampled()函数，从原图像中采样，复制部分图像，重新生成图像，语法如下：

```
bool imagecopyresampled ( resource dst_image, resource src_image, int dst_x, int dst_y, int src_x, int src_y, int dst_w, int dst_h, int src_w, int src_h )
```

imagecopyresampled()函数将一幅图像中的一块正方形区域复制到另一个图像中，平滑地插入像素值，在减

小图像大小的同时，仍然保持了极大的清晰度。

参数说明：

dst_image：目标图像标识符；

src_image：源图像标识符。

如果源和目标的宽度和高度不同，则会进行相应的图像收缩和拉伸。坐标指的是左上角。本函数可用来在同一幅图内部复制（如果 dst_image 和 src_image 相同的话）区域，但如果区域交迭的话则结果不可预知。

如果成功则返回 TRUE，失败则返回 FALSE。

注意：本函数需要 GD 库 2.0.1 版或者更高版本的支持。

（4）为重新生成的图像添加水印，可以选择文字水印或者图像水印。

添加文字水印应用 ImageTTFText()函数，如果页面使用的不是 utf-8 编码，那么还要对水印文字的编码格式进行转换，因为 ImageTTFText()函数只支持 utf-8 编码。

添加图像水印应用 imagecopy()函数，完成对图像的复制操作。

设计过程

缩略图艺术库沿用了第 4 章中实例 229 通过 header 函数进行下载的内容，在此基础上增加上传图片生成缩略图的功能，有关上传和下载这里不再赘述。

（1）在上传文件的处理页 index_ok.php 中，包含了一个 thumbnail.php 文件；重新定义上传文件的类型，并重新指定上传文件的存储位置，定义原始文件存储文件夹和缩略图存储文件夹；在通过 move_uploaded_file()函数完成文件的上传操作后，调用 thumbnail.php 文件中包含的方法 makethumb()将上传的图片生成缩略图，并添加文字水印。这是在 index_ok.php 文件中所作的处理，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include "conn/conn.php";
include "thumbnail.php";
if ($_POST [files] != "") {
$data = date ( "Y-m-d H:m:s" );
function check($var) {                                                  //验证数组的返回值是否为空
	return ($var != "");
}
$files = array_filter ( $_POST ["files"], "check" );                    //去除数组中空值
$array = array_filter ( $_FILES ["picture"] ["name"], "check" );        //去除数组中空值
foreach ( $array as $key => $value ) {                                  //循环读取数组中数据
	$types = strtolower ( strstr ( $value, '.' ) );                     //截取上传文件的后缀
	if ($types == ".jpg" || $types == ".gif" || $types == ".png") {     //判断上传文件的后缀是否符合要求
		$thumbnail = 'thumbnail/';
		$original = 'original/';
		$file_name = time () . $key . strtolower ( strstr ( $value, "." ) );   //定义上传文件名称
		$path = $original . $file_name;                                 //定义原始文件的存储位置
		$path_thumbnail = $thumbnail . $file_name;                      //定义缩略图的存储位置
		move_uploaded_file ( $_FILES ["picture"] ["tmp_name"] [$key], $path );   //执行上传操作
		makethumb ( $path, $path_thumbnail, "200", "200", "100", "www.mrbccd.com" );   //生成缩略图
		$query = "insert into tb_up_file (file_test,path_thumbnail,data,file_name) values ('$path','$path_thumbnail','$data','$files[$key]')";
		$result = mysql_query ( $query );
	} else {
		echo "上传文件" . $value . "类型不正确！ ";
	}
}
echo "<script>alert('图片上传成功'); window.location.href='index.html';</script>";
}
?>
```

（2）编写 thumbnail.php 文件，定义 makethumb()方法，完成缩略图的生成和水印的添加操作。自定义方法 makethumb()的语法如下：

```
makethumb($srcFile, $dstFile, $dstW, $dstH, $rate = 100, $markwords = null, $markimage = null){}
```

参数说明：

$srcFile：图片文件名；

$dstFile：另存文件名；

$markwords：水印文字；

$markimage：水印图片；

$dstW：图片保存宽度；

$dstH：图片保存高度；

$rate：图片保存品质，默认值为 100。

有关 thumbnail.php 文件的完成代码请参考本书光盘。

秘笈心法

心法领悟 278：定义缩略图的好处。

运用缩略图技术，可解决图片输出时的变形问题，无论是否定义输出图片的大小，它都将以一个相同的尺寸进行输出，使页面更加规范、美观，同时也避免了输出图片的拉伸变形。

实例 279	提取图像的 EXIF 信息 光盘位置：光盘\MR\06\279	高级 趣味指数：★★★★☆

实例说明

EXIF 信息是指由数码相机在拍摄过程中采集一系列的信息，然后把信息放置在 JPEG/TIFF 文件的头部，也就是说 EXIF 信息是镶嵌在 JPEG/TIFF 图像文件格式内的一组拍摄参数，主要包括摄影时的光圈、快门、ISO、日期时间等各种与当时摄影条件相关的讯息，以及相机品牌型号、色彩编码、拍摄时录制的声音和全球定位系统（GPS）等信息。既然在 EXIF 中包含了如此详细和完整的信息，那么在 PHP 中应该如何获取到这些信息呢？这就是本实例将要介绍的技术——获取并输出图像的 EXIF 信息，其运行结果如图 6.10 所示。



图 6.10　提取图像的 EXIF 信息

关键技术

在 PHP 中，获取 EXIF 信息应用的是 EXIF 函数库中的函数。首先，应该确定 EXIF 函数是否已经加载，在 php.ini 文件中查看 extension=php_exif.dll 前是否有分号“;”，如果有则说明未加载，要将分号去掉。

同时还要注意 extension=php_exif.dll 项的存储位置，必须放置在 extension=php_mbstring.dll 项之后，否则

EXIF 函数是不被支持的。设置完成后保存 php.ini 文件，并重新启动 Apache 服务器。在本实例中，具体应用的是 EXIF 函数库中的 exif_read_data()函数，获取图像的 EXIF 信息。

exif_read_data()函数，从 JPEG 或 TIFF 文件中读取 EXIF 头信息，其语法如下：

```
array exif_read_data ( string filename [, string sections [, bool arrays [, bool thumbnail]]] )
```

参数说明：

filename：被读取的图像文件名，不可以是 URL。

sections：定义存在于文件中的逗号分隔的区段列表，用来产生结果数组。如果未找到所请求的区段则返回值为 FALSE。所包含的区段如表 6.1 所示。

表 6.1　exif_read_data()函数中参数 sections 包含的区段

区段名称	包含内容、说明
FILE	FileName、FileSize、FileDateTime、SectionsFound
COMPUTED	html、Width、Height、IsColor、可能有更多其他的。Height 和 Width 是用和 getimagesize() 一样的方法计算的，因此它们的值不能是任何返回的头信息的部分；html 是一个 height/width 文本字符串，可以用于普通的 HTML 中
ANY_TAG	任何包含有标记的信息，例如 IFD0、EXIF 等
IFD0	所有 IFD0 的标记数据。在标准的图像文件中包含了图像大小及其他
THUMBNAIL	如果有第二个 IFD，文件应该包含有缩略图。所有有关嵌入缩略图的标记信息都存储在本区
COMMENT	JPEG 图像的注释头信息
EXIF	EXIF 区段是 IFDO 的子区，包含有图像的更多详细信息。大多数内容都是数码相机相关的

arrays：指定是否每个区段都成为一个数组。sections COMPUTED、THUMBNAIL 和 COMMENT 区段总是成为数组，因为它们里面包含的名字和其他区段冲突。

thumbnail：如果该值为 TRUE，读取缩略图本身；否则只读取标记数据。

exif_read_data()函数的返回值是一个关联数组，键名是头信息名，值是与其对应的数据。如果没有可供返回的数据，将返回 FALSE。

设计过程

本实例沿用缩略图艺术库中的内容，实现多图片的上传和分页输出，并且为上传的图片生成缩略图。在分页输出上传图片时，运行自定义函数 GetImageInfo()获取图片的 EXIF 信息，并且输出图片的名称和拍摄时间。

（1）在 index.php 页面中，实现上传图片的分页输出，并且包含 exif.php 文件，调用自定义函数 GetImageInfo()获取图片的 EXIF 信息，输出图片的名称和拍摄时间，其关键代码如下：

```
<?php
include "exif.php";                          //包含 exif.php 文件
$pic=GetImageInfo($myrow[file_test]);        //调用 getimageinfo()方法，获取原始图片的 EXIF 信息
echo "文件名：".$pic['文件名']."<br>";        //输出图片名称
echo " 拍摄时间：".$pic['拍摄时间'];          //输出拍摄时间
?>
```

（2）创建 exif.php 文件。编写自定义函数 getimageinfo()，应用 exif_read_data()函数获取指定图片的 EXIF 信息，其关键代码如下：

```
<?php
function GetImageInfo($img) {
        $imgtype = array ("", "GIF", "JPG", "PNG", "SWF", "PSD", "BMP" );
        $Orientation = array ("", "top left side", "top right side", "bottom right side" );
        $ResolutionUnit = array ("", "", "英寸", "厘米" );
        $exif = exif_read_data ( $img, "IFD0" );
        if ($exif === false) {
                $new_img_info = array ("文件信息" => "没有图片 EXIF 信息" );
```

```
        } else {
                $exif = exif_read_data ( $img, 0, true );
                $new_img_info = array ("文件名" => $exif [FILE] [FileName], "时间戳" => date ( "Y-m-d H:i:s", $exif [FILE] [FileDateTime] );
        }
        return $new_img_info;
}
?>
```

有关本实例的其他内容，由于篇幅所限，请参考本书光盘，这里不再赘述。

秘笈心法

心法领悟 279：通过表单中的隐藏域控制上传文件的大小。

在对上传的文件进行判断时，如果应用$_FILES['picture']['error']对上传信息进行判断，当其返回值为 2 时，说明上传文件的大小超出表单中约定的范围，而表单中的这个约定值是通过隐藏域 MAX_FILE_SIZE 的值来控制的。在设置这个隐藏域时，必须将隐藏域放置在文件域之前，否则不起作用。

6.2 控制、显示图片特性

在网页中适当地应用一些动态图片特效，不仅可以使整个页面具有更大的视觉冲击力，还可以使页面具有温馨感，最终达到提高网站访问量的目的。

实例 280	通过鼠标滑轮控制图片大小 光盘位置：光盘\MR\06\280	初级 趣味指数：★★☆

实例说明

为了合理利用网页空间，在网页设计时会缩小某些图片的实际尺寸，但这样可能导致浏览者看不清图片的内容，为了解决上述问题，可以通过鼠标滑轮改变图片的大小。运行本实例，将显示如图 6.11 所示的页面，将鼠标指针放到图片上，通过滚动鼠标滑轮即可改变图片的大小。



图 6.11　通过鼠标滑轮控制图片大小

关键技术

本实例主要通过调用自定义函数 change_img1()和自定义函数 change_img2()实现图片的大小改变，并通过 onmousewheel 事件调用这两个函数。

onmousewheel 事件用于调用 handlerText 所代表的 Javascript 方法，语法如下：

```
onmousewheel="handlerText"
```

设计过程

（1）在页面中插入两副图片，并通过这两副图片的 onmousewheel 事件调用函数 change_img1()和函数

change_img2()，代码如下：

```
<img id="img" src="images/image.gif" width="60" height="60" onmousewheel="change_img1(this)"/>
<img id="img" src="images/image.gif" width="60" height="60" onmousewheel="change_img2(this)"/>
```

（2）编写函数 change_img1()和函数 change_img2()改变图片的大小，代码如下：

```
<script language="javascript">
function change_img1(x) {
    x.width= x.width+10;
    x.height=x.height+10;
  if(x.width==1000||x.height==1000) {
      x.width=60;
x.height=60;
    }
}
function change_img2(x) {
    x.width= x.width-10;
    x.height=x.height-10;
  if(x.width==0||x.height==0) {
      x.width=60;
      x.height=60;
    }
}
</script>
```

秘笈心法

心法领悟 280：图片特效的扩展。

本实例介绍的是通过鼠标滑轮控制图片的大小，依据此原理，还可以实现图片放大镜的功能，通过倍数来控制图片的大小。

实例 281	显示随机图像 光盘位置：光盘\MR\06\281	中级 趣味指数：★★★★☆

实例说明

在用户登录页面或留言发表页面经常会看到验证码的身影，通过这项技术可以很大程度提高网站的安全性。运行本实例，可以发现在如图 6.12 所示的登录页面中随机产生了 4 位数字（6191），每次刷新页面这 4 位数字都会发生改变，用户登录时必须输入这 4 位数字，这样可以防止用户通过恶意程序来试探登录密码的值从而非法登录网站。



图 6.12　显示随机图像

关键技术

应用 PHP 的随机函数 mt_rand()生成随机验证码，根据随机验证码的值读取对应的数字图片，生成验证码图像。

mt_rand()函数，用于产生 min 和 max 之间的随机数，语法如下：

```
int mt_rand([int min],[int max])
```

参数 min 和 max 指定随机数的取值范围。

在读取生成的随机验证码时，应用到 for 语句和 substr()函数对验证码的值进行读取，根据读取的值定义输出的数字图像，其关键代码如下：

```
<?php
	$num = intval ( mt_rand ( 1000, 9999 ) );	//生成随机验证码
	for($i = 0; $i < 4; $i ++) {	//循环读取验证码
		echo "<img src=images/code/" . substr ( strval ( $num ), $i, 1 ) . ".gif>";	//输出数字图像
	}
?>
```

设计过程

（1）创建用户登录页面，添加用户登录表单。

（2）创建 JavaScript 脚本，定义 chkuserinput()方法对表单中提交的数据进行验证。

```
<script language="javascript">
function chkuserinput(form){
	if(form.username.value==""){
		alert("请输入用户名!");
		form.username.select();
		return(false);
	}
	if(form.userpwd.value=="") {
		alert("请输入用户密码!");
		form.userpwd.select();
		return(false);
	}
	if(form.yz.value==""){
		alert("请输入验证码!");
		form.yz.select();
		return(false);
	}
	return(true);
}
</script>
```

（3）编写 PHP 脚本，通过 mt_rand()函数生成 4 位随机验证码，并输出 4 位随机验证码对应的图片。

秘笈心法

心法领悟 281：mt_rand()函数的应用。

通过 mt_rand()函数获取随机数不但可以用于验证码的生成，还可以应用到随机抽奖或者随机抽取考题的程序中。其应用的原理都是相同的，如果读者感兴趣可以亲自体验一下。

实例 282　获取页面中图像的实际尺寸

中级

光盘位置：光盘\MR\06\282

趣味指数：★★★★☆

实例说明

在开发 Web 项目中，经常需要准确地获取图像的尺寸，以达到精确定位的目的。运行本实例，将显示如图 6.13 所示的页面，生成该页面的同时程序会计算出页面中图像的高度和宽度，以便于开发人员实现对图片的定位。



图 6.13　获取图像的实际尺寸

关键技术

PHP 中获取图像的大小应用 getimagesize()函数。该函数用于获取图片的实际尺寸，其语法如下：

```
array getimagesize ( string filename [, array imageinfo])
```

getimagesize()函数返回一个具有 4 个单元的数组，包括：

☑ 索引 0：图像宽度的像素值；
☑ 索引 1：图像高度的像素值；
☑ 索引 2：图像类型的标记：1 = GIF，2 = JPG，3 = PNG，4 = SWF，5 = PSD，6 = BMP，7=TIFF(intel byte order)，8=TIFF(motorola byte order)，9 = JPC，10 = JP2，11 = JPX，12 = JB2，13 = SWC，14 = IFF，15 = WBMP，16 = XBM。
☑ 索引 3：文本字符串，内容为 height="yyy" width="xxx"，可直接用于<img> 标记。

设计过程

本实例主要利用 getimagesize()函数获取图片的实际尺寸，代码如下：

```
<?php
$array=getimagesize("./mingrisoft.jpg");
?>
```

秘笈心法

心法领悟 282：getimagesize()函数的妙用。

虽然本实例只是对 getimagesize()函数的语法、基本功能进行了讲解，也许并未发现它什么特殊之处，功能也不过如此，但是当你再次回顾前面讲解的缩略图艺术库和提取图像的 EXIF 信息两个实例时，你就会发现这个函数的重要性。

实例 283	图像的手动播放 光盘位置：光盘\MR\06\283	中级 趣味指数：★★★★☆

实例说明

在图片信息较多的网页中，通过图像手动播放的方式显示图片，不仅可以节省页面空间，而且可以使浏览者自主选择喜爱的图片进行显示。运行本实例，浏览者可以通过如图 6.14 中所示的“左”和“右”文字超链接浏览所有图片。



图 6.14　获取图像的实际尺寸

关键技术

本实例主要应用 onclick()事件调用 JavaScript 脚本中的方法，完成图像的手动播放操作，在 JavaScript 中定义了 toleft()和 toright()两个方法，其代码如下：

```
<script language="javascript">
```

```
function toleft(){
        if(demo1.offsetWidth-demo.scrollLeft<=0)
                demo.scrollLeft=0
else
        demo.scrollLeft=demo.scrollLeft+10;
        }
 function toright(){
       if(demo1.offsetWidth-demo.scrollLeft<=0)
 demo.scrollLeft=0
else
             demo.scrollLeft=demo.scrollLeft-10;
}
</script>
```

设计过程

（1）设计网页的页面，定义<div>标签和插入手动播放的图像。

（2）编程控制图像播放的 JavaScript 脚本。

（3）在超链接中通过 onclick()事件完成对 JavaScript 脚本中方法的调用，其关键代码如下：

```
<td width="100"><div align="left"><a href="#" onclick="toleft()">左</a></div></td>
<td width="100"><div align="right"><a href="#" onclick="toright()">右</a></div></td>
```

秘笈心法

心法领悟 283：图像手动播放的好处。

虽然图像的手动播放没有图像自动播放灵活，但是，手动播放的最大优点是可以定位网页中播放的内容，而自动播放却不能对其进行定位，只能不停地循环播放。

实例 284	图像的自动播放 光盘位置：光盘\MR\06\284	中级 趣味指数：★★★★☆

实例说明

图像的自动播放不仅可以增添 Web 页面的动态效果，而且可以节省网页空间，有效地保证在有限的页面中显示更多的图片。运行本实例，如图 6.15 所示，图像将从右向左自动播放。



图 6.15　图像的自动播放

关键技术

本实例主要应用 setInterval()函数，在指定的时间间隔 time 内调用函数方法，完成图像的自动播放操作。setInterval()函数的语法如下：

```
setInterval(function,time)
```

设计过程

（1）设计网页的页面，定义<div>标签和插入手动播放的图像，其代码如下：

```
<div id="demo" style="OVERFLOW: hidden; WIDTH: 740px; HEIGHT: 176px" >
<table cellspacing="0" cellpadding="0" align="center" border="0">
        <tr>
         <td id="demo1"> 
<img   height="175" src="images/1.bmp" width="143" border="0" /> 
                  </td>
              <td id="demo2"></td>
        </tr>
</table>
</div>
```

（2）编程控制图像自动播放的 JavaScript 脚本，并应用 setInterval()函数完成对 Marquee()方法的调用，其代码如下：

```
<SCRIPT language="javascript">
var speed=20
demo2.innerHTML=demo1.innerHTML
function Marquee(){
if(demo1.offsetWidth-demo.scrollLeft<=0)
             demo.scrollLeft=0
else
demo.scrollLeft++
}
var MyMar=setInterval(Marquee,speed)
demo.onmouseover=function(){clearInterval(MyMar)}
demo.onmouseout=function() {MyMar=setInterval(Marquee,speed)}
</SCRIPT>
```

秘笈心法

心法领悟 284：图像自动播放的控制。

对于图像的自动播放，我们可以不了解其具体是如何运作的，只要明确它调用的方法，定义<div>标签的 ID 以及传递正确的时间间隔，即可完成图像的自动播放功能。

实例 285	任意调整上传图片的大小 光盘位置：光盘\MR\06\285	中级 趣味指数：★★★★☆

实例说明

图像大小的调整对于专业的美工或者网页设计者来说不算什么，但是对于程序员来说，特别是那些不善于处理图片的程序员来说，是一个非常头疼的问题。虽说图片处理不在程序员的工作范围之内，但是有些小问题，还是自己解决更好。在本实例中，向程序员介绍一种方法，通过我们编写的程序实现对图像大小的任意调整。运行本实例，在上传图片的右侧，选择要调整图片的大小比例，然后单击“调整图像大小”按钮，运行结果如图 6.16 所示。

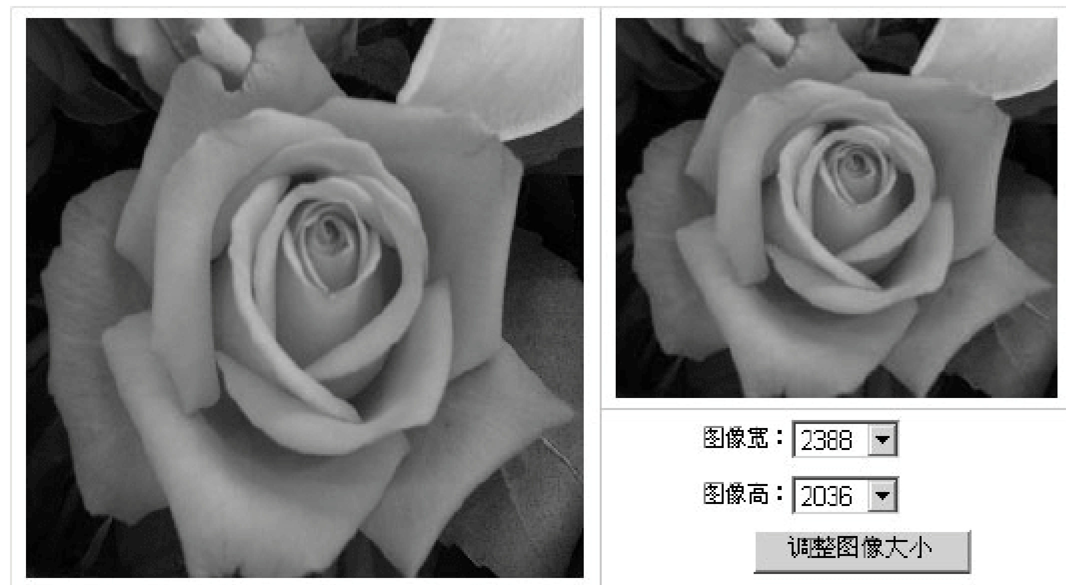


图 6.16　任意调整图像的大小

关键技术

任意调整上传图片大小方法实现的原理：首先，根据表单中提交的值，获取指定图片的存储路径和名称，以及它要被调整的比例。然后，应用 getimagesize()函数获取指定图片的数据。接着，定义 thumb()方法，根据原始图片的数据和提交的调整比例数据，判断是根据比例缩放，还是进行裁剪。最后，载入原始图片，并定义新的缩略图，应用 ImageCopyResampled()函数将原始图像复制到新的缩略图中，并将新的图像存储到指定位置。

imagecopyresampled()函数，从原图像中采样，复制部分图像，重新生成图像，详情请参见实例 278。

imagecopyresized()函数，同样实现图像的复制操作，语法如下：

```
bool imagecopyresized ( resource dst_image, resource src_image, int dst_x, int dst_y, int src_x, int src_y, int dst_w, int dst_h, int src_w, int src_h )
```

imagecopyresized()函数将一幅图像中的一块正方形区域复制到另一个图像中。

参数说明：

dst_image：目标图像标识符；

src_image：源图像标识符。

如果源和目标的宽度和高度不同，则会进行相应的图像收缩和拉伸。坐标指的是左上角。本函数可用来在同一幅图内部复制（如果 dst_image 和 src_image 相同的话）区域，但如果区域交迭的话则结果不可预知。

注意： 因为调色板图像的限制（255+1 种颜色），存在一个问题。重新采样或过滤图像通常需要多于 255 种颜色，计算新的被重新采样的像素及其颜色时采用了一种近似值。尝试为调色板图像分配一个新颜色时，如果失败那么选择计算结果最接近（理论上）的颜色（并不是视觉上最接近的颜色）。这可能会产生怪异的结果，例如空白（或者视觉上是空白）的图像。要解决这个问题，可使用真彩色图像作为目标图像，例如使用 imagecreatetruecolor()函数创建的图像。

设计过程

（1）创建一个图片上传功能，将图片上传到服务器指定的文件夹下，并通过文件系统函数读取存储在指定目录下的图片。

（2）在图片输出的同时，在右侧创建一个表单，提交图片的存储位置、名称和要调整的尺寸，将数据提交到 index_ok.php 文件中。

（3）创建 index_ok.php 文件，包含 thumbnail.php 文件，实例化 image 类，调用其中的 thumb()方法，以表单中提交的数据作为参数，完成对图片大小的调整，其关键代码如下：

```
<?php
if($_POST['Submit']=="调整图像大小"){
include("thumbnail.php");                                      //包含图像类存储文件
$img = new image();                                            //实例化图像处理类
$return=$img->param($_POST['pic_path'])->thumb("./thumbnail/".$_POST['pic_name'], $_POST['pic_width'],$_POST['pic_height'],1);
echo "<script>alert('图像大小设置成功！'); window.location.href='index.php';</script>";
}
?>
```

（4）创建 thumbnail.php 文件，编写 image 类，定义 thumb()方法，根据表单中提交的数据，完成图像大小的调整操作。其关键代码如下：

```
//thumb(新图地址，宽，高，裁剪，允许放大)
public function thumb($filename, $news_w = 100, $news_h = 100, $cut = 0, $big = 0) {
        $info = $this->getImageInfo ( $this->img );                    //获取原图信息
        if (! empty ( $info [0] )) {
                $old_w = $info [0];
                $old_h = $info [1];
                $new_w = $old_w * $news_w;
                $new_h = $old_h * $news_h;
                $type = $info ['type'];
                $ext = $info ['ext'];
                unset ( $info );
```

```
        if ($old_w < $new_h && $old_h < $new_w && ! $big) {             //如果原图比缩略图小，并且不允许放大
            return false;
        }
        if ($cut == 1) {                                                 //居中裁剪
            $scale1 = round ( $new_w / $new_h, 2 );
            $scale2 = round ( $old_w / $old_h, 2 );
            if ($scale1 > $scale2) {
                $end_h = round ( $old_w / $scale1, 2 );
                $start_h = ($old_h - $end_h) / 2;
                $start_w = 0;
                $end_w = $old_w;
            } else {
                $end_w = round ( $old_h * $scale1, 2 );
                $start_w = ($old_w - $end_w) / 2;
                $start_h = 0;
                $end_h = $old_h;
            }
            $width = $new_w;                                             //定义图像宽
            $height = $new_h;                                            //定义图像高
        }

        $createFun = 'ImageCreateFrom' . $type;                          //新建图像
        $oldimg = $createFun ( $this->img );
        if ($type != 'gif' && function_exists ( 'imagecreatetruecolor' )) {
            $newimg = imagecreatetruecolor ( $width, $height );          //创建新的真彩图像
        } else {
            $newimg = imagecreate ( $width, $height );                   //创建新的图像
        }
        if (function_exists ( "ImageCopyResampled" )) {
            ImageCopyResampled ( $newimg, $oldimg, 0, 0, $start_w, $start_h, $width, $height, $end_w, $end_h );     //完成图像的复制
        } else {
            ImageCopyResized ( $newimg, $oldimg, 0, 0, $start_w, $start_h, $width, $height, $end_w, $end_h );
        }
        $type == 'jpeg' && imageinterlace ( $newimg, 1 );                //对.jpeg 图形设置隔行扫描
        $imageFun = 'image' . $type;                                     //生成图片
        ! @$imageFun ( $newimg, $filename ) && die ( '保存失败!' );
        ImageDestroy ( $newimg );                                        //销毁图像
        return $filename;                                                //返回图像名称
    }
}
```

秘笈心法

心法领悟 285：自制图像处理工具。

在本实例中我们编写的只是一个可以按比例任意调整图像大小的方法，如果读者感兴趣可以对这个方法进行扩展，例如，向图像中添加水印、以数值的形式调整图像大小、向图像中添加文字以及给图像生成缩略图等，如果将这些功能综合到一起，那么它就变成了一个简单、实用的图像处理工具了。本实例虽然没有将这些功能进行整合，但是这里提到的每种方法，在我们的实例中都有体现，读者可以尝试自己将它们整合到一起。

实例 286　Apache 防盗链技术

光盘位置：光盘\MR\06\286

中级　趣味指数：★★★☆

实例说明

网站的图片被盗链，是一件让人非常郁闷的事情。盗链不仅盗用图片，更直接的问题是用户在下载盗链人网站上的图片时，会给服务器带来压力，导致日志中的访问记录暴涨，而带宽被耗尽。为了打击盗链行为，本节介绍一种方法，让盗链者的某些小伎俩不能得逞——APACHE 防盗技术。

对 APACHE 服务器进行设置后，当盗链者盗链网站的图片时，将输出如图 6.17 所示的效果。



图 6.17　被盗链图片和盗链后的输出结果

关键技术

要防止网站图片被盗链，就有必要先了解盗链的手段，基本可以分为以下两种：

☑　盗链者直接使用图片链接引用网站上的图片。

这种方式就是我们在实例说明中叙述的那种情况，这种盗链行为是可以完全防止的。

防止这种盗链情况的发生，有两种可行的方法：通过 Apache 服务器进行控制和使用 SESSION 变量在程序中进行控制。

☑　盗链者直接从网站上下载图片并且复制使用。

这种方式不可能被真正地阻止，我们能做的只是让它的下载更加困难而已。防止的方法有屏蔽页面中的右键使用功能、禁止对网页的下载等。

Apache 防盗技术的原理与 PHP 伪静态技术的原理是相同的，都需要应用 Apache 的 mod_rewrite.so 模组。Apache 服务器的配置文件 httpd.conf 的修改方法如下：

（1）打开 httpd.conf 文件，定位到如下位置：

```
#LoadModule rewrite_module modules/mod_rewrite.so
```

将该项前面的“#”去掉，并启动该项。

（2）查找 httpd.conf 文件，找到其中的 AllowOverride 项，将它的值都修改为 All。保存并重新启动 Apache 服务器，使修改生效。

（3）在实例根目录下创建.htaccess 文件，定义防止网站图片被盗链的方法。.htaccess 文件的代码如下：

```
SetEnvIfNoCase Referer "^http://192.168.1.59/" local_ref=1
<FilesMatch ".(gif|jpg)">
Order Allow,Deny
Allow from env=local_ref
</FilesMatch>
```

☑　Referer 字段：当 Apache 处理一个请求时，将检测头信息里的 Referer 字段，并且设置环境变量 local_ref 为 1，如果请求从本身的网站地址开始，即是本网站的一个页面。

☑　^http://192.168.1.59/：是一个正则表达式，为了设置环境变量，Referer 值必须与其匹配。

☑　"NoCase"指令：定义正则表达式的值，忽略字符串的大小写。

☑　Order Allow,Deny：设置 Apache 对当前的请求，将执行列表中的 Allow 指令，然后重复进行 Deny 指令。

☑　local_ref：设置了 local_ref 环境变量（无论什么值）的请求，而任何其他的请求将被拒绝，因为它们不符合 Allow 的条件并且默认是拒绝访问的。

这就是通过 Apache 服务器来防止图片被盗链的方法。有关通过 SESSION 变量防止图片被盗链的方法将在实例 287 中进行讲解。

设计过程

这里沿用实例 285 的内容，并在其基础上演示如何通过 Apache 防止网站图片被盗链。

（1）修改 Apache 服务器的配置文件 httpd.conf，具体步骤请参考关键技术。

（2）在实例根目录下创建.htaccess 文件，编写通过 Apache 服务器防止图片被盗链的方法，具体代码请参

考关键技术。

（3）在另外一个 Apache 服务器上重新创建一个程序，在程序中盗链本实例中的一个图片时，将出现实例说明中所示的运行结果。

秘笈心法

心法领悟 286：在 Apache 服务器的 httpd.conf 文件中定义防止图片被盗链的方法 。

在本实例中，防止图片被盗链的方法只对本实例中的图片有效，如果将.htaccess 文件中的方法定义到 httpd.conf 中，那么这个方法将对本服务器下的所有程序都有效。

实例 287	通过 SESSION 变量防盗链 光盘位置：光盘\MR\06\287	中级 趣味指数：★★★★☆

实例说明

本实例将介绍另外一种网站图片防盗链方法，即在程序中通过 SESSION 变量防止图片被盗链。当网站中应用了 SESSION 防盗链技术后，如果网站的图片被盗链，将输出如图 6.18 所示的效果。



图 6.18　盗链后的输出结果

关键技术

通过 SESSION 防盗链与通过 SESSION 屏蔽页面刷新对计数器的影响原理相同。在页面中，首先初始化 SESSION 变量，定义一个 SESSION 变量并赋值为 TRUE。然后，在插入图像的<img>标签中，定义 src 属性的值时，不直接定义指定图片的位置，而是链接到指定的.php 文件中，并且将要插入的图像名称作为超链接的参数值传递到.php 文件中。最后，在.php 文件中，通过判断 SESSION 变量的值是否为真，决定超链接中传递的图片是否被输出，而完成对网页中图片的保护。

设计过程

（1）设计一个简单的网页。首先，在这个页面中初始化一个 SESSION 变量，并且设置变量的值为 TRUE。然后，在<img>标签中，将 src 的属性值指定到 getimage.php 文件中，并且将插入的图像名称设置为超链接的参数值，其关键代码如下：

```
<?php
session_start();                                                    //初始化 SESSION 变量
$_SESSION['viewimages']=true;                                       //定义 SESSION 变量，并为其赋值
?>
<tr>
        <td colspan="2">
                <img src="getimage.php?img=bg3_01.jpg" width="267" height="31" alt=""></td>
        <td>
                <img src="getimage.php?img=bg3_02.jpg" width="536" height="31" alt=""></td>
</tr>
```

（2）创建 getimage.php 文件，在该文件中通过对 SESSION 变量值的判断，决定是否输出图像，其关键代码如下：

```
<?php
session_start();                                                    //初始化 SESSION 变量
if (isset($_SESSION['viewimages']) && $_SESSION['viewimages'] == true) {    //判断 SESSION 变量是否为真
 if (isset ( $_GET ['img'] )) {                                     //超链接传递的值是否存在
        $dims = getimagesize ( 'images/' . $_GET ['img'] );         //获取超链接传递图片的信息
        header ( 'Content-Disposition: inline; filename=' . $_GET ['img'] );
        header ( 'Content-Type:' . $dims ['mime'] );
        header ( 'Content-Length:' . filesize ( 'images/' . $_GET ['img'] ) );
```

```
        readfile ( 'images/' . $_GET ['img'] );                                    //完成对图片的读取操作
} else {
        header ( "HTTP/1.1 404 NOT Found" );
        header ( "Content-Type: text/plain" );
        echo "图像不存在！ \n";
}
} else {
header ( "HTTP/1.1 404 NOT Found" );
header ( "Content-Type: text/plain" );
echo "这是一个被保护的图像不可盗链！ \n";
}
```

秘笈心法

心法领悟 287：防止图片被盗链方法的选择。

通过 SESSION 变量防盗链相对通过 Apache 防盗链来说麻烦一些，但是如果要在购买的虚拟主机中使用，那么必须考虑虚拟主机的 Apache 配置是否支持 mod_rewrite.so 模组。而使用 SESSION 变量来防止图片被盗链，则不必考虑这个问题，因为它完全是在用户的程序中运行。所以，具体使用哪种方法来防止网站图片被盗链，可以根据实际情况而定。

6.3　Jpgraph 类库

Jpgraph 是基于 GD2 函数库编写的主要用于创建统计图的类库。在绘制统计图方面不仅功能非常强大，而且代码编写方便，只需简单的几行代码就可以绘制出非常复杂的统计图效果，从而很大程度地提高了编程人员的开发效率。在本节中，将通过设计各种类型的统计图，让大家全面地体会 Jpgraph 类库的强大功能。

实例 288	柱形图分析产品月销售量 光盘位置：光盘\MR\06\288	初级 趣味指数：★★★

实例说明

在本实例中，将介绍如何通过 Jpgraph 类库创建柱形图，完成对产品月销售量的统计分析，其运行效果如图 6.19 所示。

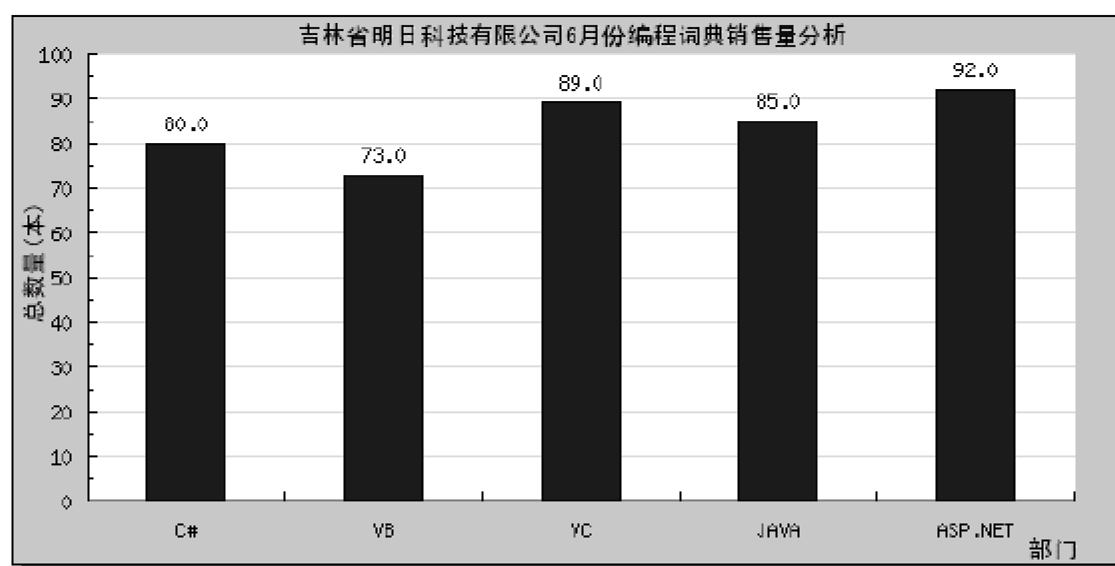


图 6.19　柱形图分析产品月销售量

关键技术

要运用 Jpgraph 类库，首先必须了解它如何下载、都有哪些版本、都适用于哪些环境。Jpgraph 包括 JpGraph 1.x 系列、JpGraph 2.x 系列和 JpGraph 3.x 系列。

☑　JpGraph 1.x 系列仅适用于 PHP 4 环境，PHP5 下不能工作。

☑　JpGraph 2.x 系列仅适用于 PHP 5 环境(≥5.1.x)，PHP 4 下不能工作。目前 2.x 系列的最新版本是 2.3.4。

☑　JpGraph 3.x 系列是 2.x 的升级版本，目前最新的版本是 3.0.6。

本书中应用的是最新版本 JpGraph 3.0.6。

配置 Jpgraph 的最简单的方法就是将解压后的 src 文件夹直接复制到范例的根目录下。这样就可以直接在程序中调用 src 文件夹下的文件，应用 Jpgraph 类库绘制统计图。

设计过程

（1）将 Jpgraph 类库导入到程序中。

使用 Jpgraph 类库，首先要从网站 http://www.aditus.nu/jpgraph/下载该类库的压缩包并解压，将呈现如图 6.20 所示的目录结构。

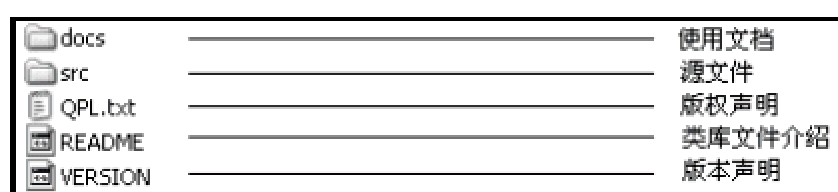


图 6.20　Jpgraph 压缩包解压后的目录

（2）将 src 文件夹复制到实例的根目录下。

（3）创建 index.php 文件，将 Jpgraph 导入到项目中，具体操作步骤如下：

☑　使用 Graph 类创建统计图对象；

☑　调用 Graph 类的 SetScale()方法设置统计图的刻度样式；

☑　调用 Graph 类的 SetShadow()方法设置统计图阴影；

☑　调用 Graph 类的 img 属性的 SetMargin()方法设置统计图的边界范围；

☑　调用 BarPlot 类创建统计图的柱形效果；

☑　调用 BarPlot 类的 SetFillColor()方法设置柱形图的前景色。

index.php 的关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require_once 'src/jpgraph.php';                                          //导入 Jpgraph 类库
require_once 'src/jpgraph_bar.php';                                      //导入 Jpgraph 类库的柱形图功能
$data = array(80, 73, 89, 85, 92);                                       //设置统计数据
$datas = array("C#", "VB", "VC", "JAVA", "ASP.NET");                     //设置统计数据
$graph = new Graph(600, 300);                                            //设置画布大小
$graph->SetScale('textlin');                                             //设置坐标刻度类型
$graph->SetShadow();                                                     //设置画布阴影
$graph->img->SetMargin(40, 30, 20, 40);                                  //设置统计图边距
$barplot = new BarPlot($data);                                           //实例化 BarPlat 对象
$barplot->SetFillColor('blue');                                          //设置柱形图前景色
$barplot->value->Show();
$graph->Add($barplot);
$graph->title->Set(iconv("utf-8","gb2312","吉林省明日科技有限公司 6 月份编程词典销售量分析"));    //统计图标题
$graph->xaxis->title->Set(iconv("utf-8","gb2312","部门"));               //设置 X 轴名称
$graph->xaxis->SetTickLabels($datas);
$graph->yaxis->title->Set(iconv("utf-8","gb2312",'总数量(本)'));          //设置 Y 轴名称
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                              //设置标题字体
$graph->xaxis->title->SetFont(FF_SIMSUN,FS_BOLD);                        //设置 X 轴字体
$graph->yaxis->title->SetFont(FF_SIMSUN,FS_BOLD);                        //设置 Y 轴字体
$graph->Stroke();                                                        //输出图像
```

秘笈心法

心法领悟 288：Jpgraph 在 Apache 服务器下的通用配置。

如果想要 Jpgraph 在 Apache 服务器下的所有站点均有效，可以按如下步骤进行配置：

（1）将解压后的 src 文件夹保存到服务器磁盘中，例如 c:\jpgraph。

（2）编辑 php.ini 文件，修改 include_path 的配置项，在该项后增加 Jpgraph 类库的保存目录，例如 include_path = ".;c:\jpgraph"。

（3）重新启动 Apache 服务，配置生效。

这样 Jpgraph 就在 Apache 服务器下的所有站点中都有效了。

实例 289	柱形图展示年度收支情况 光盘位置：光盘\MR\06\289	高级 趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成柱形图，对公司年度总的收支情况进行分析，其运行结果如图 6.21 所示。

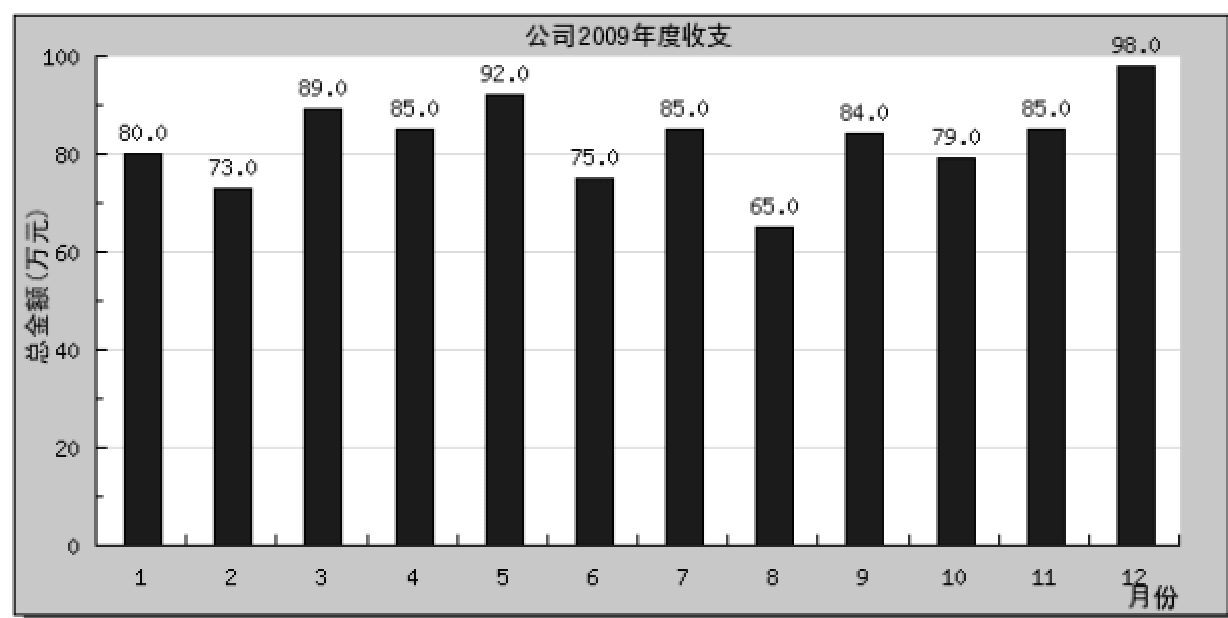


图 6.21　公司年度收支展示

关键技术

在本实例中以年为单位，分析在一年中公司每个月的收支情况。首先，确立一年中每个月的业绩。然后，确定柱形图的标题、X 轴、Y 轴显示的内容。最后，将 src/jpgraph.php 导入到项目中，完成柱形图的创建，其具体方法解析如下：

（1）通过 include()语句包含创建指定图像所需的文件。在本实例中创建柱形图，需要使用的文件如下：

```
require_once '../src/jpgraph.php';                    //导入 Jpgraph 类库
require_once '../src/jpgraph_bar.php';                //导入 Jpgraph 类库的柱形图功能
```

（2）使用 Graph 类创建统计图对象。

使用 Jpgraph 类库首先需要创建统计图对象，统计图对象使用 Graph 类创建，该类的使用语法如下：

```
$graph = new Graph($w, $h);
```

参数说明：

$w：指定图片的宽度。

$h：指定图片的高度。

（3）调用 Graph 类的 SetScale()方法设置统计图的刻度样式。

统计图的 X 轴和 Y 轴的样式有多种，可以使用 Graph 类的 SetScale()方法设置统计图刻度的样式，SetScale()方法的语法如下：

```
$graph->SetScale($type)
```

$type：表示 X 轴样式和 Y 轴样式的组合，具体组合方式如表 6.2 所示。

表 6.2　SetScale()方法 X 轴与 Y 轴样式说明

坐　标　轴	直 线 样 式	文 本 样 式	对 数 样 式	整 数 样 式
X	Lin	text	Log	Int
Y	Lin	无	Log	Int

例如，textlog 表示 X 轴坐标为文本样式，Y 轴为对数样式的 X-Y 刻度。

（4）调用 Graph 类的 SetShadow()方法设置统计图阴影。

默认情况，应用 Jpgraph 绘制出统计图的下边框和右边框是没有阴影效果的，使用 Graph 类的 SetShadow()方法可以为统计图添加阴影效果，从而使统计图具有立体感。SetShadow()方法的语法如下：

```
$graph->SetShadow()
```

（5）调用 Graph 类的 img 属性的 SetMargin()方法设置统计图的边界范围。

使用 SetMargin()方法可以为统计图设置边界范围，使统计图制作变得更为灵活，该方法的语法如下：

```
$graph->img->SetMargin($left, $right, $up, $bottom)
```

参数说明：

$left：统计图左边距。

$right：统计图右边距。

$up：统计图上边距。

$bottom：统计图下边距。

（6）BarPlot 类创建统计图的柱形效果。

使用 Jpgraph 创建柱形图需要使用 BarPlot 类，该类的构造方法中包含一个数值数组参数，该数组的内容即为需要统计的数据。BarPlot 类的语法如下：

```
$barPlot = new BarPlot($data)
```

$data：由统计数据所组成的数组。

（7）调用 BarPlot 类的 SetFillColor()方法设置柱形图的前景色。SetFillColor()方法的语法如下：

```
$barplot->SetFillColor($color);
```

$color：要设置的颜色值。

设计过程

（1）创建 index.php 文件，设置页面的编码格式，将 src 文件夹导入到实例根目录下。

```
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
require_once '../src/jpgraph.php';                                   //导入 Jpgraph 类库
require_once '../src/jpgraph_bar.php';                               //导入 Jpgraph 类库的柱形图功能
```

（2）应用 Jpgraph 类库中的方法完成柱形图的创建，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
require_once 'src/jpgraph.php';                                      //导入 Jpgraph 类库
require_once 'src/jpgraph_bar.php';                                  //导入 Jpgraph 类库的柱形图功能
$data = array(80, 73, 89, 85, 92, 75 ,85 ,65, 84, 79, 85, 98);       //设置统计数据
$graph = new Graph(600, 300);                                        //设置画布大小
$graph->SetScale('textlin');                                         //设置坐标刻度类型
$graph->SetShadow();                                                 //设置画布阴影
$graph->img->SetMargin(40, 30, 20, 40);                              //设置统计图边距
$barplot = new BarPlot($data);                                       //实例化 BarPlat 对象
$barplot->SetFillColor('blue');                                      //设置柱形图前景色
$barplot->value->Show();
$graph->Add($barplot);
$graph->title->Set(iconv("utf-8","gb2312",'公司 2009 年度收支'));        //设置统计图标题
$graph->xaxis->title->Set(iconv("utf-8","gb2312",'月份'));             //设置 X 轴名称
$graph->yaxis->title->Set(iconv("utf-8","gb2312",'总金额(万元)'));      //设置 Y 轴名称
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                          //设置标题字体
$graph->xaxis->title->SetFont(FF_SIMSUN,FS_BOLD);                    //设置 X 轴字体
```

```
$graph->yaxis->title->SetFont(FF_SIMSUN,FS_BOLD);                //设置 Y 轴字体
$graph->Stroke();                                                //输出图像
```

秘笈心法

心法领悟 289：转换编码格式的重要性。

由于本实例采用的是 utf-8 编码，所以在生成的图像中输出中文字符串时，必须对其进行编码转换，同时要设置字体为 SetFont(FF_SIMSUN, FS_BOLD)，如果使用其他类型的字体，即使完成编码转换，输出的中文字符串仍然是乱码。

实例 290 柱形图展示编程词典 6、7 月份销售量

光盘位置：光盘\MR\06\290

高级

趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成对比柱形图，对编程词典 6、7 月份的销售量进行比较，其运行结果如图 6.22 所示。

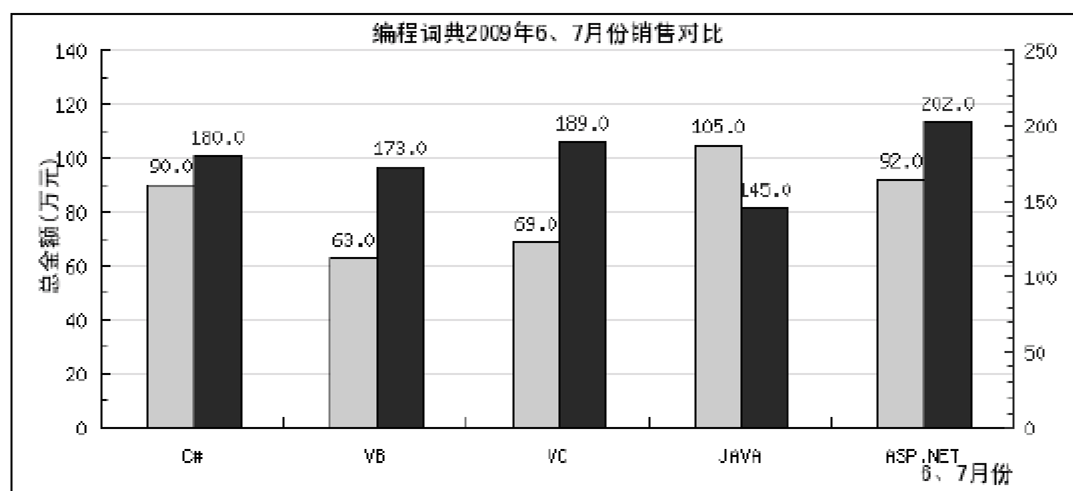


图 6.22　编程词典 6、7 月份销量对比

关键技术

在本实例中通过柱形图完成对编程词典 6、7 月份销量的对比。此例在单一柱形图的基础上增加一组柱形图，形成两组数据的对比效果，其实现方法与单一柱形图是相同的，可以说是创建两个单一的柱形图，然后将其在同一图像中输出。

（1）通过 BarPlot 类创建柱形图，运用其返回对象调用 value()方法中的 Show()方法输出数据，通过 SetColor()方法设置输出数据颜色。

SetColor()方法的语法如下：

```
SetColor($aColor, $aLabelColor)
```

参数说明：

$aColor：设置坐标轴的颜色；

$aLabelColor：设置坐标轴上标签的颜色，默认值为 false。

（2）通过 Add()方法将柱形图数据添加到图像中，其参数是柱形图创建返回的对象，语法如下：

```
Graph ::
Add($aPlot)
```

$aPlot：指定图像创建时返回的对象。

（3）在本实例中，还应用到 GroupBarPlot 类创建柱形图数组，通过该类可以实现对柱形图的分段处理，如果重新设置数组$datazero 的值，那么柱形图会发生一些变化。

```
Class GroupBarPlot Extends BarPlot
GroupBarPlot ::
GroupBarPlot($plots)
```

GroupBarPlot 创建一个新的图像对象，$plots 是一个数组。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法完成对比柱形图的创建，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include ("../src/jpgraph.php");                                          //导入文件
include ("../src/jpgraph_bar.php");                                      //导入文件
$data_july = array(90, 63, 69, 105, 92);                                 //设置统计数据
$data_june = array(180, 173, 189, 145, 202);                             //设置统计数据
$datazero=array(0,0,0,0,0);
$datas = array("C#", "VB", "VC", "JAVA", "ASP.NET");                    //设置统计数据
$graph = new Graph(600,270);                                             //创建图像
$graph->SetScale("textlin");                                             //设置坐标刻度类型
$graph->SetY2Scale("lin");
$graph->img->SetMargin(50, 40, 20, 40);                                  //设置统计图边距
$graph->yaxis->scale->SetGrace(20);
$graph->y2axis->scale->SetGrace(20);
$graph->SetMarginColor('white');                                         //创建背景颜色
$graph->y2axis->SetColor('darkred');
$bplotzero = new BarPlot($datazero);                                     //设置图像的填充值
$ybplot1 = new BarPlot($data_july);                                      //创建 6 月份数据
$ybplot1->value->Show();
$ybplot = new GroupBarPlot(array($ybplot1,$bplotzero));
$ybplot2 = new BarPlot($data_june);                                      //创建 7 月份数据
$ybplot2->value->Show();                                                 //输出数据
$ybplot2->value->SetColor('darkred');
$ybplot2->SetFillColor('darkred');
$y2bplot = new GroupBarPlot(array($bplotzero,$ybplot2));
//将数据添加到图像中
$graph->Add($ybplot);
$graph->AddY2($y2bplot);
$graph->title->Set(iconv("utf-8","gb2312",'编程词典 2009 年 6、7 月份销售对比'));    //设置统计图标题
$graph->xaxis->title->Set(iconv("utf-8","gb2312",'6、7 月份'));            //设置 X 轴名称
$graph->xaxis->SetTickLabels($datas);
$graph->yaxis->title->Set(iconv("utf-8","gb2312",'总金额(万元)'));          //设置 Y 轴名称
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                              //设置标题字体
$graph->xaxis->title->SetFont(FF_SIMSUN,FS_BOLD);                        //设置 X 轴字体
$graph->yaxis->title->SetFont(FF_SIMSUN,FS_BOLD);                        //设置 Y 轴字体
$graph->Stroke();                                                        //生成图像
?>
```

秘笈心法

心法领悟 290：字体设置的重要性。

在通过 Jpgraph 类库创建图像时，要想在图像中输出中文字符串，只对字符串的编码格式进行转换是不够的，还要为中文字符串设置正确的字体（SetFont(FF_SIMSUN, FS_BOLD)），否则无论怎么转换字符串的编码格式都是徒劳的。

实例 291	柱形图展示编程词典上半年销量 光盘位置：光盘\MR\06\291	高级 趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成柱形图，对公司编程词典上半年的销量进行统计，其运行结果如图 6.23 所示。

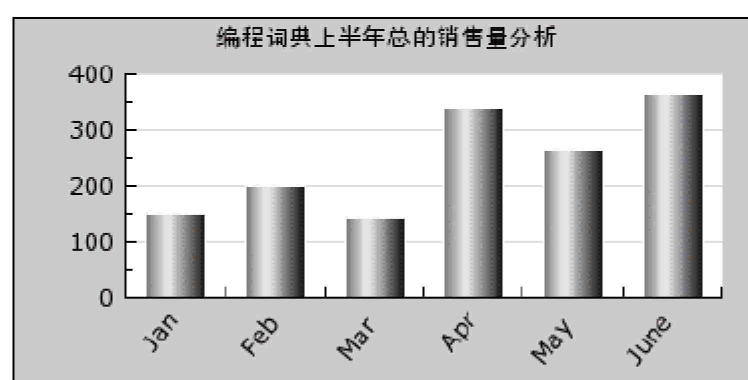


图 6.23　编程词典上半年销量展示

关键技术

在本实例中，对编程词典上半年的销量进行分析，这里创建的柱形图与上面实例创建的柱形图的唯一区别是形状和颜色不同。

本实例通过 BarPlot 类中的 SetFillGradient()方法完成对柱形图颜色和形状的设置，其语法如下：

```
SetFillGradient($aFromColor, $aToColor, $aStyle)
```

参数说明：

$aFromColor：默认颜色；

$aToColor：转换的颜色；

$aStyle：设置的样式。

本实例中的具体应用如下：

```
SetFillGradient ("navy","#FFFF00",GRAD_LEFT_REFLECTION);  //设置图像的类型和填充颜色
```

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法完成特殊柱形图的创建，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include ("../src/jpgraph.php");
include ("../src/jpgraph_bar.php");
$datay=array(150,201,145,340,265,365);
$datax=array("Jan","Feb","Mar","Apr","May","June");
$graph = new Graph(400,200,"auto");                                     //创建图像
$graph->img->SetMargin(60,20,30,50);                                    //设置图像边框间距
$graph->SetScale("textlin");                                            //定义坐标刻度类型
$graph->SetMarginColor("lightblue");                                    //定义图像颜色
$graph->title->Set(iconv("utf-8","gb2312",'编程词典上半年总的销售量分析'));    //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                             //设置标题字体
$graph->xaxis->SetFont(FF_VERDANA,FS_NORMAL,10);                        //设置 X 轴的字体
$graph->yaxis->SetFont(FF_VERDANA,FS_NORMAL,10);                        //设置 Y 轴的字体
$graph->xaxis->SetTickLabels($datax);                                   //设置 X 轴输出的数据
$graph->xaxis->SetLabelAngle(50);                                       //设置输出文字大小
$bplot = new BarPlot($datay);                                           //实例化图像创建类
$bplot->SetWidth(0.6);                                                  //设置柱形图的输出大小
$bplot->SetFillGradient("navy","#FFFF00",GRAD_LEFT_REFLECTION);         //设置图像的类型和填充颜色
$bplot->SetColor("white");                                              //设置图像边框颜色
$graph->Add($bplot);                                                    //添加数据
$graph->Stroke();                                                       //生成图像
?>
```

秘笈心法

心法领悟 291：SetFillGradient()方法的妙用。

通过 SetFillGradient()方法不但可以改变柱形图的形状，还可以使柱形图变得更加光滑，其方法如下：

```
//$bplot->SetFillGradient("navy","lightsteelblue",GRAD_MIDVER);
```

实例 292　柱形图展示 2009 年上半年总销售额

光盘位置：光盘\MR\06\292

高级

趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成柱形图，对公司年度总的收支情况进行分析，其运行结果如图 6.24 所示。

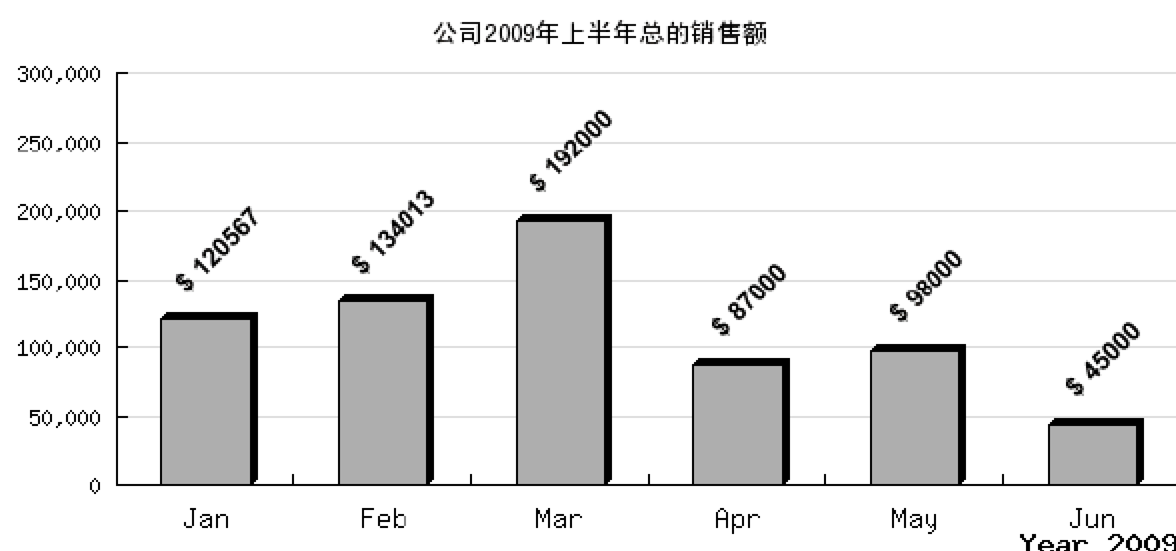


图 6.24　公司 2009 年上半年总销售额

关键技术

在本实例中，通过柱形图展示公司 2009 年上半年总的销售额，这个柱形图的特点体现在背景图像、X 轴时间、数据输出和数据的格式化上。

（1）通过图像的 SetFrame()方法控制背景图像，其语法如下：

```
SetFrame($aDrawImgFrame, $aImgFrameColor, $aImgFrameWeight)
```

参数说明：

$aDrawImgFrame：如果将该参数设置为 FALSE，那么就不输出背景图像；如果设置为 TRUE 或者不应用该方法那么将输出背景图像。

$aImgFrameColor：该参数是一个数组，定义背景图像的颜色。

$aImgFrameWeight：默认值为 1，设置背景宽度。

（2）X 轴上时间的设置。大家可以看到，X 轴上的时间坐标采用的是月份的英文简写格式，这个时间数据不是通过数组定义的，而是应用 GetShortMonth()方法直接生成的，其代码如下：

```
$a = $gDateLocale->GetShortMonth();                //定义 X 轴输出的内容，英文月份的简写
```

然后通过图像中 xaxis 对象中的 SetTickLabels()方法将数据添加到 X 轴上，其代码如下：

```
$graph->xaxis->SetTickLabels($a);                  //添加数据
```

（3）柱形图中数据的输出，应用的是柱形图 value 对象中的 Show()方法，然后应用 SetAngle()方法设置数据的输出角度，应用 SetFormat()方法对数据进行格式化，其关键代码如下：

```
$bplot->value->Show();                             //输出图像对应的数据值
$bplot->value->SetAngle(45);                       //定义图像值的输出角度
$bplot->value->SetFormat('$ %0.0f');               //对输出值进行格式化
```

（4）调用自定义函数，对数据进行格式化操作。本实例应用图像中 yaxis 对象中的 SetLabelFormatCallback()方法调用自定义函数，执行数据的格式化操作，其关键代码如下：

```
$graph->yaxis->SetLabelFormatCallback('yScaleCallback');    //执行数字的格式化输出
```

其中 yScaleCallback 为自定义函数的名称。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的

Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个没有背景图像的柱形图，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include ("../src/jpgraph.php");
include ("../src/jpgraph_bar.php");
function yScaleCallback($aVal) {                                     //定义数字格式化的方法
        return number_format($aVal);
}
$datay=array(120567,134013,192000,87000,98000,45000);
$graph = new Graph(650,280,'auto');                                  //创建图像
$graph->img->SetMargin(80,30,30,40);                                 //定义图像边框间距
$graph->SetScale("textint");                                         //定义刻度值的类型
$graph->SetShadow();                                                 //设置图像阴影
$graph->SetFrame(false);                                             //不设置图像的背景
$graph->yaxis->scale->SetGrace(35);                                  //设置 Y 轴距离顶部的距离
$graph->yaxis->SetLabelFormatCallback('yScaleCallback');             //执行数字的格式化输出
$a = $gDateLocale->GetShortMonth();                                  //定义 X 轴输出的内容，英文月份的简写
$graph->xaxis->SetTickLabels($a);                                    //添加数据
$graph->xaxis->SetFont(FF_FONT2);                                    //设置字体
$graph->title->Set(iconv("utf-8","gb2312",'公司 2009 年上半年总的销售额'));   //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                          //设置标题字体
$graph->xaxis->title->Set("Year 2009");                              //设置 X 轴的角标
$graph->xaxis->title->SetFont(FF_FONT2,FS_BOLD);                     //定义字体
$bplot = new BarPlot($datay);                                        //创建图像
$bplot->SetFillColor("orange");                                      //定义图像的填充颜色
$bplot->SetWidth(0.5);                                               //设置图像的大小
$bplot->SetShadow();                                                 //设置图像的阴影
$bplot->value->Show();                                               //输出图像对应的数据值
$bplot->value->SetFont(FF_ARIAL,FS_BOLD);                            //定义图像值的字体
$bplot->value->SetAngle(45);                                         //定义图像值的输出角度
$bplot->value->SetFormat('$ %0.0f');                                 //对输出值进行格式
$bplot->value->SetColor("black","darkred");                          //定义值的颜色
$graph->Add($bplot);                                                 //添加数据
$graph->Stroke();                                                    //生成图像
?>
```

秘笈心法

心法领悟 292：通过 Jpgraph 创建图像的技巧。

在通过 Jpgraph 类库创建图像时，可以在 src\Examples 文件夹下选择你想要创建的图像样式，然后向这个样式中添加属于自己的数据，如果感觉不满意，可以对大小、颜色等进行重新调整。这是通过 Jpgraph 创建图像最简单、最直接的方法。

实例 293	柱形图展示 2009 年第一季度编程词典销量 光盘位置：光盘\MR\06\293	高级 趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成柱形图，对编程词典 2009 年第一季度的销量情况进行分析，其运行结果如图 6.25 所示。

关键技术

在本实例中，通过多柱形图展示 2009 年第一季度编程词典的销量，其中包括 C#、VB、VC、JAVA 和 ASP.NET5 个语言编程词典的销量。

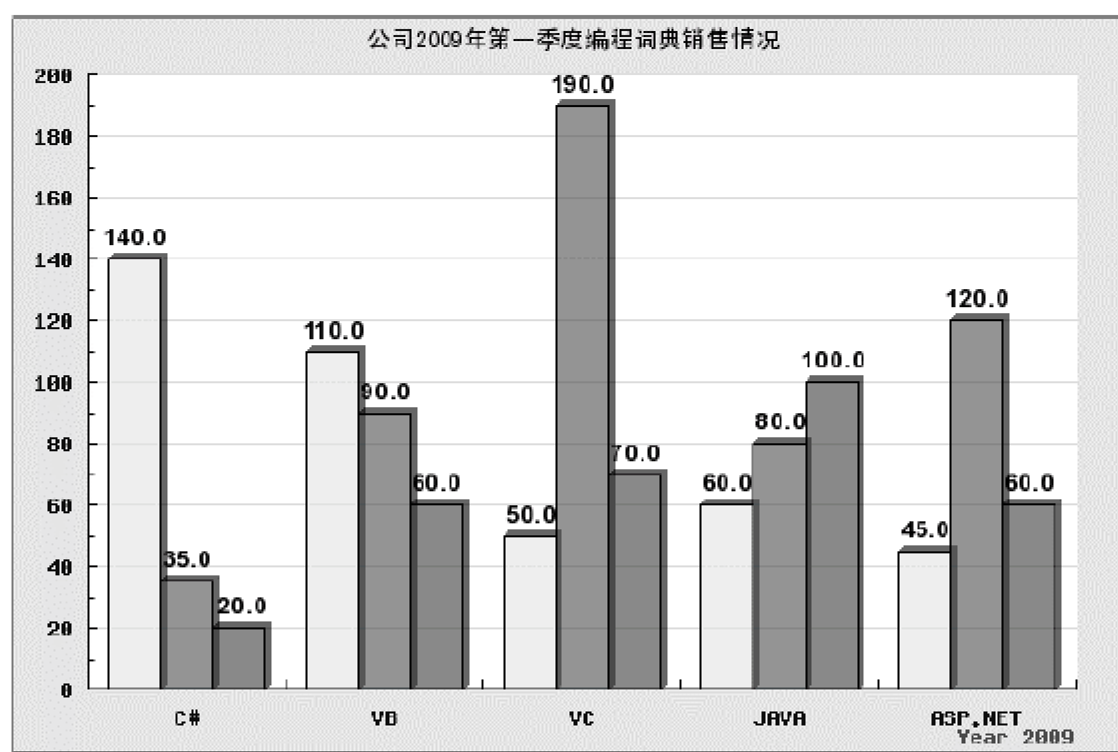


图 6.25　2009 年第一季度编程词典销量展示

本实例中柱形图创建体现的是多柱形图直接对比输出，其关键是将 BarPlot 类创建的单柱形图通过 GroupBarPlot 类整合到同一个图像中进行输出。

（1）BarPlot 类创建统计图的柱形效果。

使用 Jpgraph 创建柱形图需要使用 BarPlot 类，该类的构造方法中包含一个数值数组参数，该数组的内容即为需要统计的数据。BarPlot 类的使用格式如下：

```
$barPlot = new BarPlot($data)
```

参数$data：由统计数据所组成的数组。

（2）GroupBarPlot 类创建多个柱形效果。

使用 Jpgraph 创建多个柱形图需要使用 GroupBarPlot 类，该类的构造方法中包含一个数组参数，该数组的内容即为需要整合的单个柱形图对象。GroupBarPlot 类的语法如下：

```
$gbarplot= new GroupBarPlot ($data)
```

参数$data：由单个柱形图对象所组成的数组。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个柱形图数组，对 2009 年第一季度编程词典销量进行分析，其关键代码如下：

```
$datay1=array(140,110,50,60,45);
$datay2=array(35,90,190,80,120);
$datay3=array(20,60,70,100,60);
$graph = new Graph(600,400);                                    //创建图像
$graph->SetScale("textlin");                                    //设置刻度值类型
$graph->SetMargin(40,30,30,40);                                 //定义边框距离
$graph->SetMarginColor('white:0.9');                            //设置图像边框颜色
$graph->SetColor('white');                                      //设置图像的填充颜色
$datas = array("C#", "VB", "VC", "JAVA", "ASP.NET");            //设置统计数据
$graph->xaxis->SetTickLabels($datas);
$graph->xaxis->title->Set('Year 2009');                         //设置 X 轴标题
$graph->xaxis->title->SetFont(FF_FONT1,FS_BOLD) ;               //设置标题字体
$graph->xaxis->title->SetColor('red');                          //设置字体颜色
$graph->xaxis->SetFont(FF_FONT1,FS_BOLD);                       //设置 X 轴字体
$graph->xaxis->SetColor('navy');                                //设置 X 轴刻度线颜色
$graph->yaxis->SetFont(FF_FONT1,FS_BOLD);                       //设置 Y 轴字体
$graph->yaxis->SetColor('navy');                                //设置 Y 轴刻度线颜色
$graph->title->Set(iconv("utf-8","gb2312",'公司 2009 年第一季度编程词典销售情况'));  //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                     //设置标题字体
$graph->title->SetMargin(3);                                    //设置图像距离顶部距离
//创建 3 个方柱统计图
```

```
$bplot1 = new BarPlot($datay1);
$bplot2 = new BarPlot($datay2);
$bplot3 = new BarPlot($datay3);
//省略了部分代码
$bplot1->value->Show();                                    //输出图像对应的数据值
$bplot1->value->SetFont(FF_ARIAL,FS_BOLD);                 //定义图像值的字体
$bplot1->value->SetColor("black","darkred");               //定义值的颜色
//省略了部分代码
//实例化图像操作组类
$gbarplot = new GroupBarPlot(array($bplot1,$bplot2,$bplot3));
$gbarplot->SetWidth(0.8);                                  //设置图像大小
$graph->Add($gbarplot);                                    //添加数据
$graph->Stroke();                                          //生成图像
```

秘笈心法

心法领悟 293：图像颜色的填充。

在 Jpgraph 类库中，应用 SetFillColor()方法为图像填充颜色；通过 SetColor()方法定义数据、文字、坐标轴的颜色。

实例 294 折线图分析网站一天内的访问走势

光盘位置：光盘\MR\06\294

高级

趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成折线图，分析网站一天内的访问走势，其运行结果如图 6.26 所示。

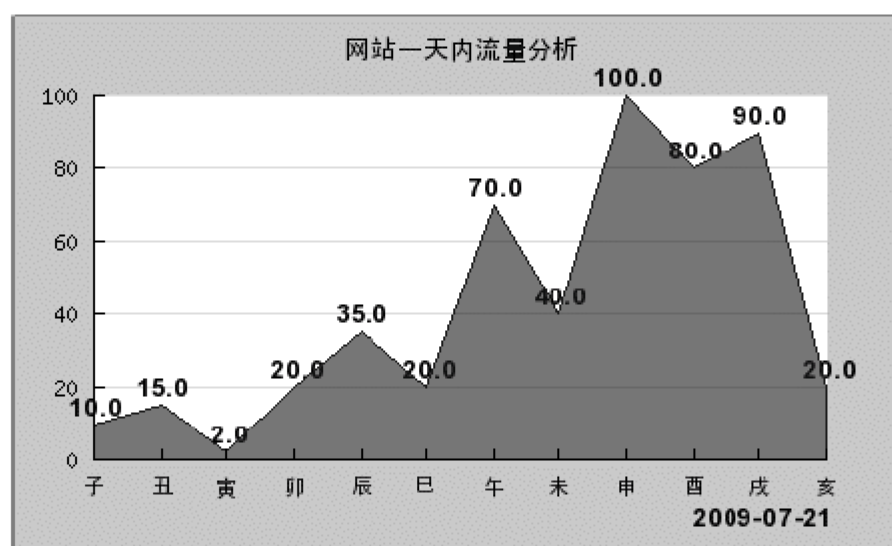


图 6.26　网站访问走势分析

关键技术

在本实例中，以一天的十二个时辰为单位，分析在这十二个时辰中网站访问量的走势。使用 Jpgraph 类库创建折线统计图，除了需要在程序中包含 jpgraph.php 文件外，还需要包含 jpgraph_line.php 文件，从而启用 Jpgraph 类库的折线创建功能。其中使用的 Jpgraph 技术如下：

（1）使用 LinePlot 对象绘制曲线。

通过 Jpgraph 类库中的 LinePlot 类创建曲线，该类的语法如下：

```
$linePlot = new LinePlot($data)                            //创建折线图
```

参数$data：数值型数组，指定统计数据。

（2）SetFont()方法统计图标题、坐标轴等文字样式。

制作统计图时，需要对图像的标题、坐标轴内文字进行样式设置，在 Jpgraph 类库中，可以使用 SetFont()实现，该方法的语法如下：

```
SetFont($family, [$style,] [$size])
```

参数说明：

$family：指定文字的字体。

$style：指定文字的样式。

$size：指定文字的大小，默认为 10。

（3）SetMargin()方法设置图像、标题、坐标轴上文字与边框的距离，其语法如下：

```
SetMargin($left,$right,$top,$bottom)
```

参数指定其与左右、上下边框的距离。

或者：

```
SetMargin($data)
```

参数$data 同样指定与边框的距离。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。注意本实例中创建的是折线图，所以导入的文件也发生了变化。

（2）应用 Jpgraph 类库中的方法创建一个折线图，对网站一天内的访问量走势进行分析，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include ("../src/jpgraph.php");
include ("../src/jpgraph_line.php");
//这里省略了创建数据的代码
$graph = new Graph(450,275);                                       //创建图像
$graph->SetMargin(40,40,40,50);                                    //设置图像的边框
$graph->SetScale("textint");                                       //定义刻度值的类型
$graph->SetShadow();                                               //设置图像阴影
$graph->title->Set(iconv("utf-8","gb2312",'网站一天内流量分析'));    //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                        //设置标题字
$graph->title->SetMargin(10);                                      //设置标题字
$graph->xaxis->SetTickLabels($datas);                              //添加 X 轴上的数据
$graph->xaxis->SetFont(FF_SIMSUN, FS_BOLD,8);                      //定义字体
$graph->xaxis->title->Set("2009-07-21");                           //设置 X 轴的角标
$graph->xaxis->title->SetFont(FF_ARIAL,FS_BOLD);                   //定义字体
$graph->xaxis->title->SetMargin(10);                               //定义位置
$pl = new LinePlot($datay);                                        //创建折线图像
$pl->value->Show();                                                //输出图像对应的数据值
$pl->value->SetFont(FF_ARIAL,FS_BOLD);                             //定义图像值的字体
$pl->value->SetColor("black","darkred");                           //定义值的颜色
$pl->SetColor("blue");                                             //定义图像颜色
$pl->SetFillColor("blue@0.4");                                     //定义填充颜色
$graph->Add($pl);                                                  //添加数据
$graph->Stroke();                                                  //输出图像
?>
```

秘笈心法

心法领悟 294：创建不同的图像导入的文件是有所区别的。

如果创建柱形图，那么导入的是：

```
include ("../src/jpgraph.php");
include ("../src/jpgraph_bar.php");
include ("../src/jpgraph_flags.php");
```

如果创建折线图，那么导入的是：

```
include ("../src/jpgraph.php");
include ("../src/jpgraph_line.php");
```

这点必须注意，如果没有导入正确的文件，就不能够完成图像的创建操作。

实例 295 柱形图与折线图分析图书销量和市场占有率

光盘位置：光盘\MR\06\295

高级

趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 实现柱形图与折线图的结合，对 2009 年图书销量和市场占有率进行分析，其运行结果如图 6.27 所示。

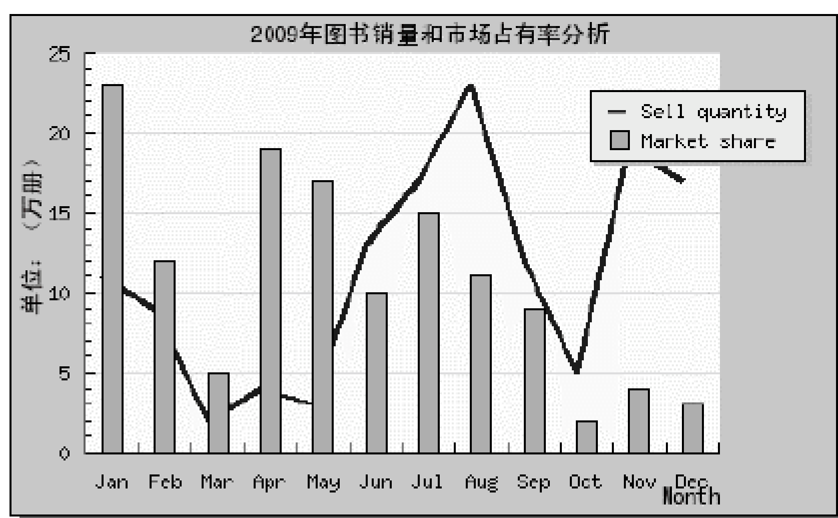


图 6.27　2009 年图书销量和市场占有率分析

关键技术

在本实例中，运用 Jpgraph 对 2009 年图书销量和市场占有率进行分析，旨在介绍如何实现柱形图与折线图的结合，同时渗入一个图像注释的应用。

柱形图与折线图的结合，就是在同一图像中分别创建一个柱形图和一个折线图，与单独创建柱形图和折线图是相同的，唯一的区别就是要同时包含柱形图文件和折线图文件。

```
include ("../src/jpgraph.php");
include ("../src/jpgraph_line.php");
include ("../src/jpgraph_error.php");
include ("../src/jpgraph_bar.php");
```

本实例中添加了一个图像注释，通过图像的颜色来体现，黄色代表图书销量数据，蓝色代表市场占有率分析数据，其具体的关键技术如下：

在创建的图像对象中，应用 SetLegend()方法添加注释的内容，其关键代码如下：

```
$l1plot->SetLegend("Sell quantity");
```

其参数就是指定要输出的注释内容。

通过 SetColor()和 SetFillColor()方法设置注释对应的图像颜色。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。注意本实例中要导入的文件包括折线图和柱形图两种。

（2）应用 Jpgraph 类库中的方法在同一图像中创建一个柱形图和一个折线图，完成对图书销量和市场占有率的分析，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include ("../src/jpgraph.php");
include ("../src/jpgraph_line.php");
include ("../src/jpgraph_error.php");
include ("../src/jpgraph_bar.php");
$l1datay = array(11,9,2,4,3,13,17,23,12,5,19,17);
$l2datay = array(23,12,5,19,17,10,15,11,9,2,4,3);
$datax=array("Jan","Feb","Mar","Apr","May","Jun","Jul","Aug","Sep","Oct","Nov","Dec");
```

```
$graph = new Graph(450,275,"auto");                              //创建图像
$graph->img->SetMargin(40,70,20,40);                             //定义图像边距
$graph->SetScale("textlin");                                     //定义刻度类型
$graph->SetColor(array(250,250,250));                            //定义颜色
$l1plot=new LinePlot($l1datay);                                  //创建图像
$l1plot->SetColor("blue");                                       //设置图像颜色
$l1plot->SetWeight(2);                                           //设置折线宽度
$l1plot->SetLegend("Sell quantity");
$l2plot = new BarPlot($l2datay);                                 //创建图像
$l2plot->SetFillColor("orange");                                 //图像填充颜色
$l2plot->SetLegend("Market share");                              //输出注释内容
$graph->Add($l1plot);                                            //向图像中添加数据
$graph->Add($l2plot);                                            //向图像中添加数据
$graph->title->Set(iconv("utf-8","gb2312","2009 年图书销量和市场占有率分析"));   //定义标题
$graph->xaxis->title->Set("Month");                              //定义 X 轴角标
$graph->yaxis->title->Set(iconv("utf-8","gb2312","单位：（万册）"));  //定义 Y 轴标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                      //定义字体
$graph->yaxis->title->SetFont(FF_SIMSUN, FS_BOLD);               //定义字体
$graph->xaxis->title->SetFont(FF_SIMSUN, FS_BOLD);               //定义字体
$graph->xaxis->SetTickLabels($datax);                            //定义 X 轴标题
$graph->Stroke();                                                //生成图像
?>
```

秘笈心法

心法领悟 295：通过 SetWeight()方法设置折线图折线的宽度。

在 Jpgraph 类库中，创建折线图时，可以通过 SetWeight()方法设置折线图折线的宽度，其语法如下：

```
$l1plot->SetWeight($data);                                       //设置折线宽度
```

$l1plot 指创建折线图返回的对象，参数$data 是折线图折线的宽度值。

实例 296 折线图分析 2009 年牛肉市场价格走势

光盘位置：光盘\MR\06\296

高级

趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成折线图，对 2009 年牛肉市场的价格走势进行分析，其运行结果如图 6.28 所示。

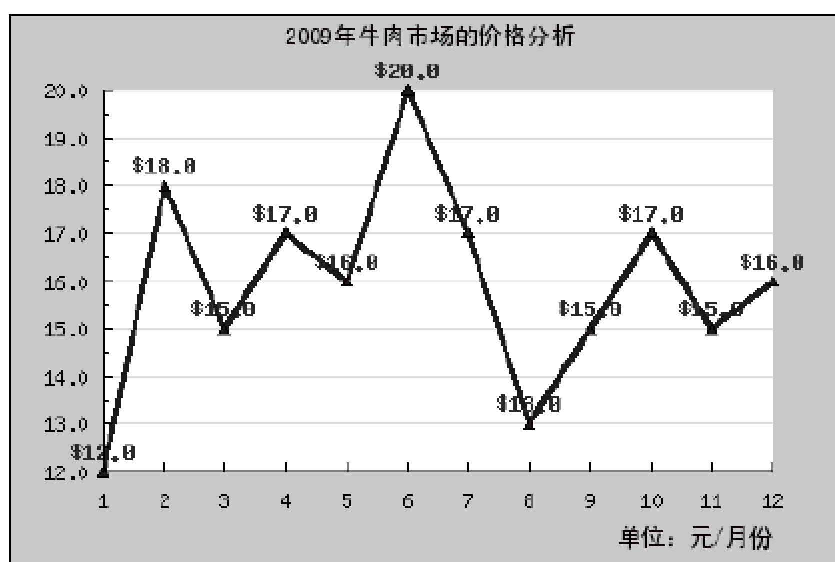


图 6.28 2009 年牛肉市场价格走势分析

关键技术

在本实例中，通过 Jpgraph 类库创建折线图，分析 2009 年牛肉市场的价格走势。旨在突出如何在折线图中输出格式化的数据，以及在折线图的数据点上添加标记。

（1）输出数据应用的是 value()对象中的 Show()方法，数据的格式化应用的是 SetFormat()方法。

```
SetFormat($aFormat, $aNegFormat)
```

参数说明：

$aFormat：设置格式化的样式；

$aNegFormat：设置 negative 的值。

（2）折线图中标记的添加应用的是图像的 mark 对象中的 SetType()方法，其参数值用于设置标记的类型，语法如下：

```
SetType($aType, $aFileName, $aScale)
```

参数说明：

$aType：设置标记的类型；

$aFileName：插入图像的名称或者国家的名称；

$aScale：设置插入标记的大小。

SetType() 方法的参数 $aType 的可选类型包括 MARK_SQUARE、MARK_UTRIANGLE、MARK_DTRIANGLE、MARK_DIAMOND、MARK_CIRCLE，这些标记仅供参考，如果想了解更多内容请参考 Jpgraph 类库的参考手册。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个折线图，对 2009 年牛肉市场价格走势进行分析，其代码如下：

```
$ydata = array(12,18,15,17,16,20,17,13,15,17,15,16);                    //定义数据
$graph = new Graph(450,300,"auto");                                     //创建图像
$graph->SetScale("textlin");                                            //定义刻度值类型
$graph->img->SetMargin(50,40,40,55);                                    //设置边框的间距
$lineplot=new LinePlot($ydata);                                         //创建折线图
$lineplot->mark->SetType(MARK_UTRIANGLE);                               //定义折线图的标记点
$lineplot->value->show();                                               //输出折线图对应的值
$lineplot->value->SetColor('darkred');                                  //设置值的颜色
$lineplot->value->SetFont(FF_FONT1,FS_BOLD);                            //设置值的字体
$lineplot->value->SetFormat('$%0.1f');                                  //对值进行格式化
$graph->Add($lineplot);                                                 //添加数据
$graph->title->Set(iconv("utf-8","gb2312","2009 年牛肉市场的价格分析"));  //定义标题
$graph->xaxis->title->Set(iconv("utf-8","gb2312","单位：元/月份"));       //定义 X 轴输出的值
$graph->xaxis->title->SetMargin(10);                                    //定义 X 轴输出的值
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                             //设置标题字体
$graph->yaxis->title->SetFont(FF_SIMSUN, FS_BOLD);                      //设置 Y 轴内容的字体
$graph->xaxis->title->SetFont(FF_SIMSUN, FS_BOLD);                      //设置 X 轴内容的字体
$lineplot->SetColor("blue");                                            //设置图像颜色
$lineplot->SetWeight(2);                                                //设置折线的宽度
$graph->Stroke();                                                       //生成图像
?>
```

秘笈心法

心法领悟 296：在 Jpgraph 类库中字体的选择。

在 Jpgraph 类库中，可以设置的字体很多。但是在选择字体时必须注意，有些字体是不支持中文的。

实例 297	折线图分析 2009 年销售额 光盘位置：光盘\MR\06\297	高级 趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成折线图，对公司 2009 年的销售额进行分析，其运行结果如图 6.29 所示。

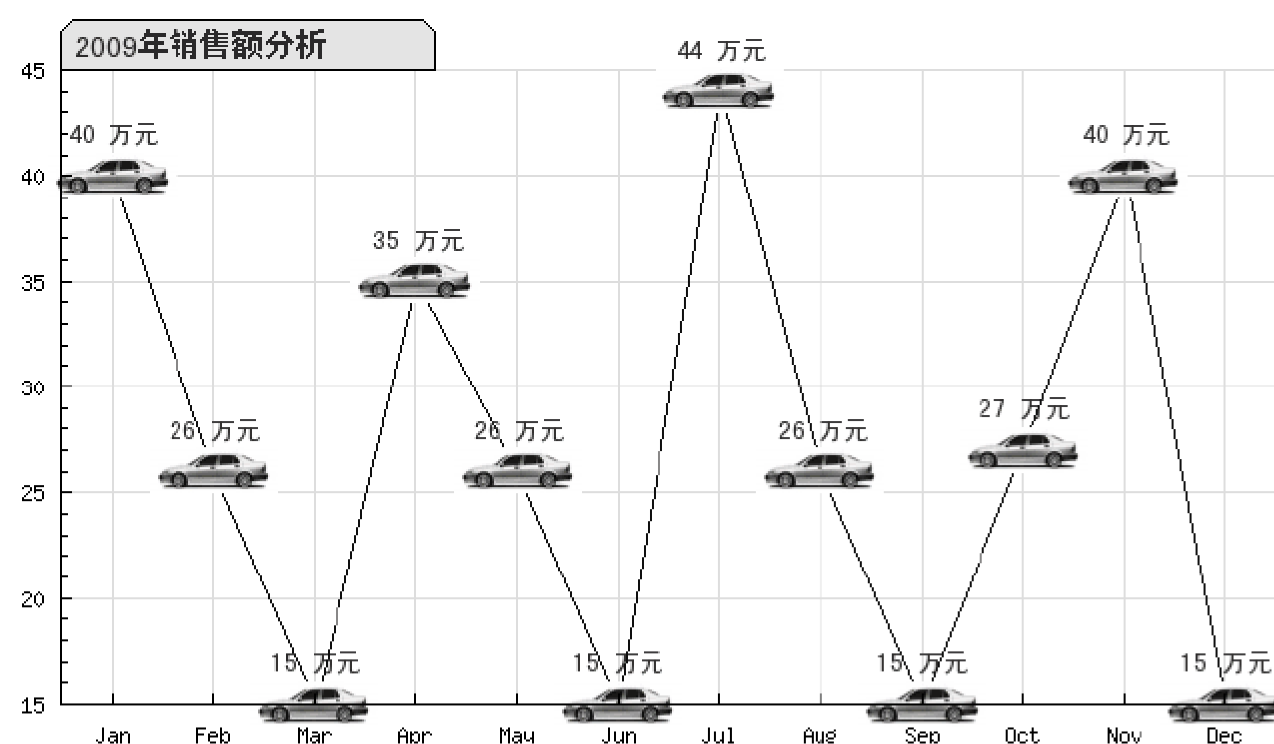


图 6.29　2009 年公司的销售额

关键技术

在本实例中，通过 Jpgraph 类库创建折线图，分析 2009 年公司的销售额。旨在突出如何在折线图中通过插入的图像设置数据点标签。

实例 296 中运行 SetType()方法的第 1 个参数，直接设置标签的样式，在本实例中，运用 SetType()方法的第 2 个参数，通过插入的图像设置数据点的标签，并通过第 3 个参数设置插入图像的大小，其语法请参考实例 296。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个折线图，对 2009 年公司的销售额进行分析，其代码如下：

```
$datay1 = array(40,26,15,35,26,15,44,26,15,27,40,15);
$graph = new Graph(650,375);                                          //定义图像大小
$graph->SetMarginColor('white');                                      //背景颜色
$graph->SetScale("textlin");                                          //定义刻度值类型
$graph->SetFrame(false);
$graph->SetMargin(30,5,35,20);                                        //设置边距
$graph->tabtitle->Set(iconv("utf-8","gb2312",'2009 年销售额分析') );  //输出标题
$graph->tabtitle->SetFont(FF_SIMSUN, FS_BOLD,12);                     //设置标题字体
$graph->tabtitle->SetColor('darkred','#E1E1FF');                      //设置标题颜色
$graph->xgrid->Show();                                                //设置阴影
$graph->xaxis->SetTickLabels($gDateLocale->GetShortMonth());          //定义 X 轴上的数据
$p1 = new LinePlot($datay1);                                          //创建图像
$p1->SetColor("navy");                                                //设置图像颜色
$p1->mark->SetType(MARK_IMG,'saab_95.jpg',0.5);                       //设置标签的样式，使用图片
$p1->value->SetFormat('%d '.iconv("utf-8","gb2312",'万元'));          //格式化数据
$p1->value->Show();                                                   //输出阴影
$p1->value->SetColor('darkred');                                      //定义颜色
$p1->value->SetFont(FF_SIMSUN, FS_BOLD,10);                           //设置字体
$p1->value->SetMargin(14);                                            //设置字体大小、位置
$p1->SetCenter();
$graph->Add($p1);                                                     //添加数据
$graph->Stroke();                                                     //生成图像
```

秘笈心法

心法领悟 297：在 Jpgraph 类库中通过旗帜来设置数据点的标签。

通过旗帜设置数据点的标签，同样应用 SetType()方法，只是需要将第 1 个参数设置为 MARK_FLAG 系列（1、2、3、4 表示旗帜的大小不同），然后将第 2 个参数设置为对应旗帜的表示即可。

实例 298 柱形图分析编程词典销售比例

光盘位置：光盘\MR\06\298

高级

趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成柱形图，对编程词典的销售比例进行分析，其运行结果如图 6.30 所示。

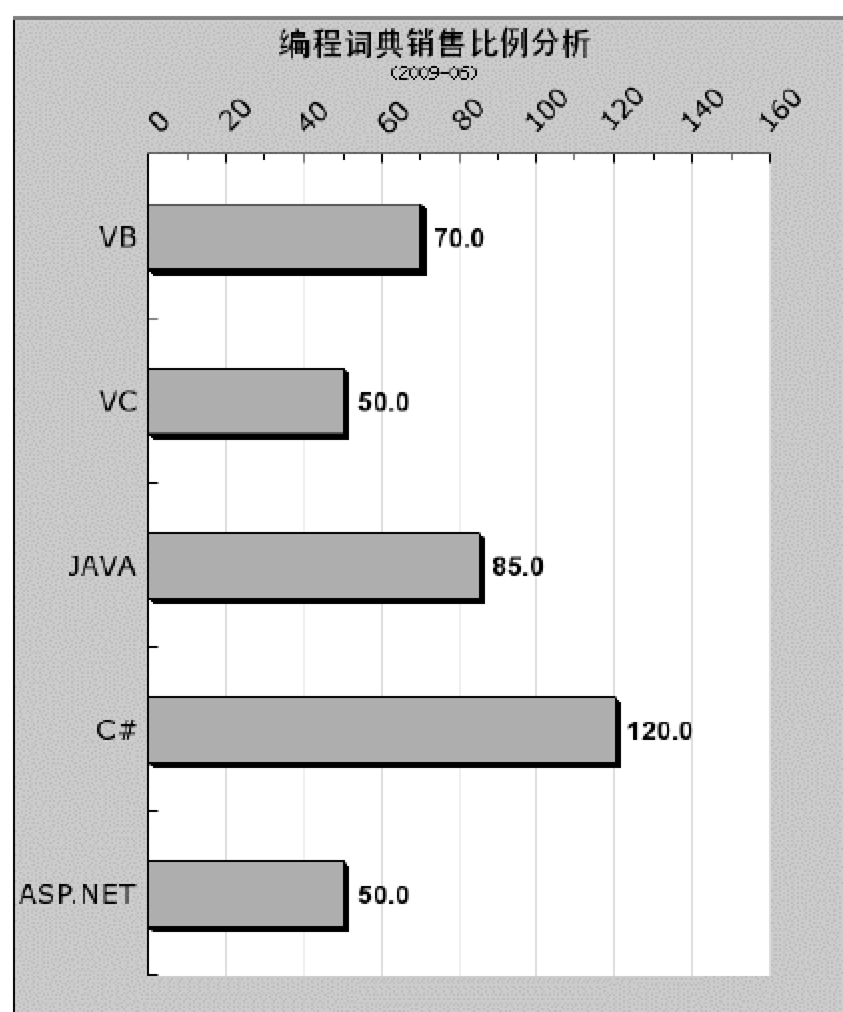


图 6.30　编程词典销售比例分析

关键技术

在本实例中，通过 Jpgraph 类库创建柱形图，分析编程词典的销售比例。本实例中再次创建柱形图，与以往的有些不同，这里将柱形图顺时针旋转了 90°。

主要应用图像对象中的 Set90AndMargin()方法，完成柱形图的顺时针 90°旋转，其语法如下：

```
Set90AndMargin($lm, $rm, $tm, $bm)
```

参数说明：

$lm：左边框旋转角度；

$rm：右边框旋转角度；

$tm：上边框旋转角度；

$bm：下边框旋转角度。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个柱形图，从另外一个角度分析编程词典的销售比例，其代码如下：

```
$datay=array(70,50,85,120,50);
$datax=array("VB","VC","JAVA","C#","ASP.NET");
$width=500;                                              //定义图像宽
$height=600;                                             //定义图像高
$graph = new Graph($width,$height,'auto');               //创建图像
$graph->SetScale("textlin");                             //定义刻度值类型
```

```
$top = 80;
$bottom = 30;
$left = 80;
$right = 50;
$graph->Set90AndMargin($left,$right,$top,$bottom);                //设置图像的边距
$graph->title->Set(iconv("utf-8","gb2312","编程词典销售比例分析"));  //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD,14);                    //设置标题字体
$graph->subtitle->Set("(2009-06)");                               //设置副标题
$graph->xaxis->SetTickLabels($datax);                             //定义 X 轴上的数据
$graph->xaxis->SetFont(FF_VERDANA,FS_NORMAL,12);                  //设置字体
$graph->xaxis->SetLabelAlign('right','center');                   //设置 Y 轴上字体右侧居中
$graph->yaxis->scale->SetGrace(20);                               //设置 Y 轴间距
$graph->yaxis->SetLabelAlign('center','bottom');                  //设置 Y 轴字体居中，底部对齐
$graph->yaxis->SetLabelAngle(45);                                 //设置字体的倾斜度
$graph->yaxis->SetLabelFormat('%d');                              //输出数据的格式
$graph->yaxis->SetFont(FF_VERDANA,FS_NORMAL,12);                  //设置字体
$bplot = new BarPlot($datay);                                     //创建图像
$bplot->SetFillColor("orange");                                   //定义图像颜色
$bplot->SetShadow();                                              //设置图像阴影
$bplot->value->Show();                                            //设置数据阴影
$bplot->value->SetFont(FF_ARIAL,FS_BOLD,12);                      //设置数据字体
$bplot->value->SetAlign('left','center');                         //设置数据位置
$bplot->value->SetColor("black","darkred");                       //设置数据颜色
$bplot->value->SetFormat('%.1f');                                 //格式化数据
$graph->Add($bplot);                                              //添加数据
$graph->Stroke();                                                 //生成图像
```

秘笈心法

心法领悟 298：在 Jpgraph 类库中控制坐标轴上标签的位置。

在 Jpgraph 类库中，控制坐标轴上标签的位置应用的是 SetLabelAlign()方法，在本实例中多次使用了这个方法，完成对数据在边框和坐标上位置的控制。

实例 299　饼形图分析 2009 年图书销量　高级

光盘位置：光盘\MR\06\299

趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成饼形图，对公司年度图书销量情况进行分析，其运行结果如图 6.31 所示。

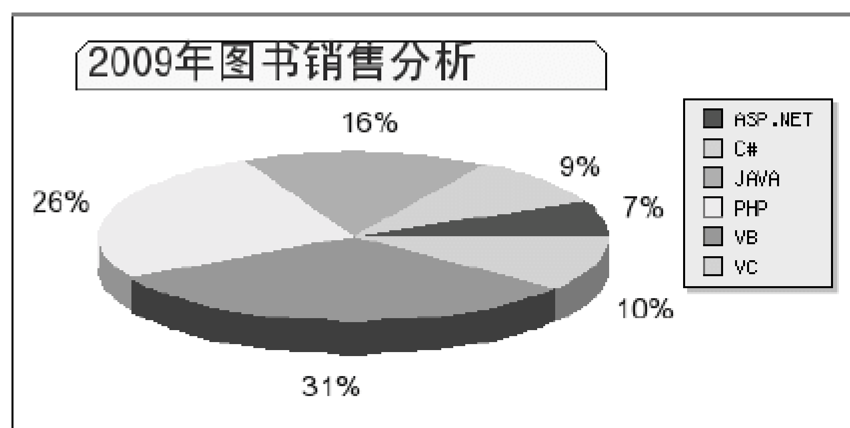


图 6.31　2009 年图书销量分析

关键技术

在本实例中，通过 Jpgraph 类库创建饼形图，分析 2009 年图书销量情况，旨在突出饼形图的创建方法，其应用的关键技术如下：

（1）创建饼形图图像应用 PieGraph 类，该类继承 Graph 类，创建 3D 饼形图的图像。

（2）创建 3D 饼形图应用 PiePlot3D 类，该类继承 PiePlot 类。

（3）通过图像中的 legend 对象调用 Pos()方法定位饼形图在图像中的存储位置。Pos()方法的语法如下：

```
Pos($aX, $aY, $aHAlign, $aVAlign)
```

参数说明：

$aX：设置距离 X 轴的位置；

$aY：设置距离 Y 轴的位置；

$aHAlign：设置右侧（居中、居左或者居右）；

$aVAlign：设置顶部（居下、居上或者居中）。

（4）通过饼形图对象调用 SetTheme()方法设置饼形图的颜色。SetTheme()方法的语法如下：

```
PiePlot ::
SetTheme($aTheme)
```

参数$aTheme 指定主题的名称，其支持的主题包括 earth、pastel、sand、water。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个饼形图，对 2009 图书销量进行分析，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include ("../src/jpgraph.php");
include ("../src/jpgraph_pie.php");
include ("../src/jpgraph_pie3d.php");
$data = array(20,27,45,75,90,30);
$graph = new PieGraph(500,275,"auto");                                  //创建图像
$graph->SetShadow();                                                    //输出阴影
$graph->tabtitle->Set(iconv("utf-8","gb2312",'2009 年图书销售分析') );  //输出标题
$graph->tabtitle->SetFont(FF_SIMSUN, FS_BOLD,18);                       //设置标题字体
$graph->title->SetColor("darkblue");                                    //定义标题颜色
$graph->legend->Pos(0.1,0.2);                                           //控制注释文字的位置
$p1 = new PiePlot3d($data);                                             //创建 3D 饼形图图像
$p1->SetTheme("water");                                                 //控制图像颜色的主题
$p1->SetCenter(0.35);                                                   //控制图像的大小比例
$p1->SetAngle(30);                                                      //控制图像的倾斜角度
$p1->value->SetFont(FF_ARIAL,FS_NORMAL,12);                             //设置字体
$p1->SetLegends(array("ASP.NET","C#","JAVA","PHP","VB","VC"));
$graph->Add($p1);                                                       //添加数据
$graph->Stroke();                                                       //生成图像
?>
```

秘笈心法

心法领悟 299：在 Jpgraph 类库中控制饼形图大小和倾斜角度的方法。

在 Jpgraph 类库中，通过饼形图对象中的 SetCenter()方法控制饼形图的大小比例，默认值是 0.5；通过 SetAngle()方法控制图像的倾斜角度，倾斜角度必须在 10°~80°之间。

实例 300	饼形图展示各语言编程词典销售比例 光盘位置：光盘\MR\06\300	高级 趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成饼形图，对公司各语言编程词典的销售比例进行分析，其运行结果如图 6.32 所示。

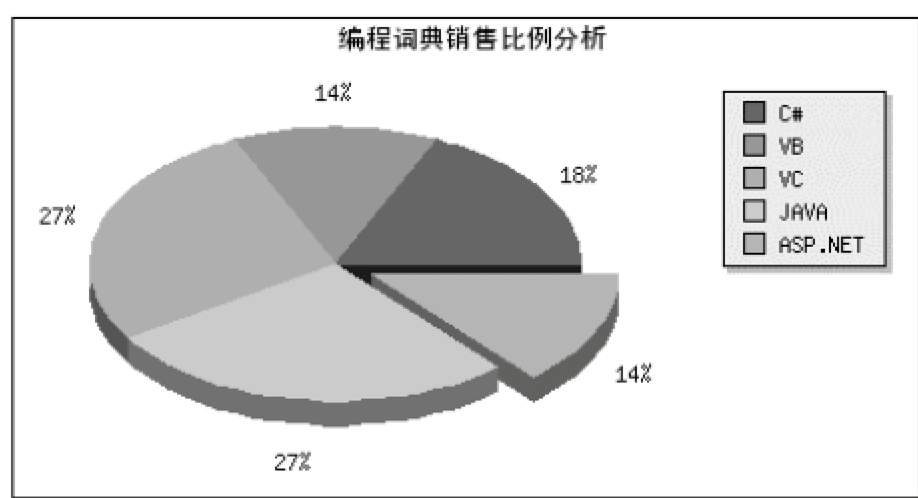


图 6.32　编程词典销售比例

关键技术

在本实例中，通过 Jpgraph 类库创建饼形图，分析 2009 年各语言编程词典的销售比例，旨在突出饼形图的创建方法，其应用的关键技术如下：

（1）创建饼形图图像应用 PieGraph 类，该类继承 Graph 类，创建 3D 饼形图的图像。

（2）创建 3D 饼形图应用 PiePlot3D 类，该类继承 PiePlot 类。

（3）通过 ExplodeSlice()方法对饼形图进行分割，其语法如下：

```
ExplodeSlice($e, $radius)
```

参数说明：

$e：指定分割的区块；

$radius：指定分割的区块与原图的距离。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个饼形图，对 2009 年各语言编程词典的销量进行分析，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include ("../src/jpgraph.php");
include ("../src/jpgraph_pie.php");
include ("../src/jpgraph_pie3d.php");
$data = array(20,15,30,30,15);
$datas = array("C#", "VB", "VC", "JAVA", "ASP.NET");                //设置统计数据
$graph = new PieGraph(450,240,"auto");                              //创建图像
$graph->SetShadow();                                                //创建图像阴影
$graph->title->Set(iconv("utf-8","gb2312","编程词典销售比例分析"));    //定义标题
$graph->title->SetFont(FF_SIMSUN, FS_BOLD);                         //设置标题字体
$p1 = new PiePlot3D($data);                                         //生成饼形图
$p1->ExplodeSlice(4,50);                                            //设置饼形图分割位置
$p1->SetCenter(0.35);                                               //设置饼形图在图像中的存储位置
$p1->SetLegends($datas);                                            //设置说明文字的内容
$graph->Add($p1);                                                   //添加数据
$graph->Stroke();                                                   //生成图像
?>
```

秘笈心法

心法领悟 300：如何保持 3D 饼形图的完整性。

在 Jpgraph 类库中创建 3D 饼形图时，如果不应用 ExplodeSlice()方法，那么创建的 3D 饼形图就是完整的；如果应用 Explode()方法，将一个数组值作为参数，那么就可以实现 3D 饼形图的多块分割。

实例 301 多饼形图区块分析 2009 年图书销量

光盘位置：光盘\MR\06\301

高级

趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 生成多饼形图区块，对公司 2009 年图书销量进行分析，其运行结果如图 6.33 所示。

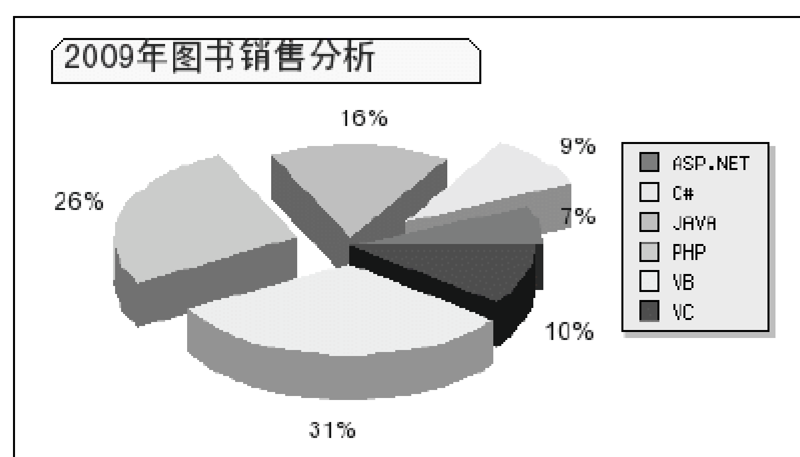


图 6.33　2009 年图书销量分析

关键技术

在本实例中，通过 Jpgraph 类库创建饼形图，应用饼形图对象中的 Explode()方法完成对 2009 年各部门图书销量的分块展示。Explode()方法的语法如下：

```
Explode($aExplodeArr)
```

该方法将饼形图分割成多个模块，参数$aExplodeArr 是一个数组，根据数组元素个数定义分割数量，根据数组元素值定义分割后模块与原图像的距离。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个饼形图，对 2009 年图书销量进行分析，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include ("../src/jpgraph.php");
include ("../src/jpgraph_pie.php");
include ("../src/jpgraph_pie3d.php");
$data = array(20,27,45,75,90,30);
$graph = new PieGraph(500,245,"auto");                                  //创建图像
$graph->SetShadow();                                                    //创建图像阴影
$graph->tabtitle->Set(iconv("utf-8","gb2312",'2009 年图书销售分析') );   //输出标题
$graph->tabtitle->SetFont(FF_SIMSUN, FS_BOLD,14);                       //设置标题字体
$graph->title->SetColor("darkblue");                                    //定义标题颜色
$graph->legend->Pos(0.1,0.2);                                           //控制注释文字的位置
$p1 = new PiePlot3d($data);                                             //创建图像
$p1->SetTheme("sand");                                                  //控制图像颜色
$p1->SetCenter(0.4);                                                    //设置图像位置
$p1->SetSize(0.4);                                                      //设置图像大小
$p1->SetHeight(20);                                                     //设置饼形图高度
$p1->SetAngle(45);                                                      //设置图像倾斜角度
$p1->Explode(array(5,40,10,30,20));                                     //控制饼形图的分割
$p1->value->SetFont(FF_ARIAL,FS_NORMAL,10);                             //设置字体
$p1->SetLegends(array("ASP.NET","C#","JAVA","PHP","VB","VC"));
$graph->Add($p1);                                                       //添加数据
$graph->Stroke();                                                       //生成图像
?>
```

秘笈心法

心法领悟 301：设置 3D 饼形图的高度。

在 Jpgraph 类库中设置 3D 饼形图的高度，应用饼形图对象中的 SetHeight()方法。

实例 302	多饼形图分析 2009 年上半年编程词典销量 光盘位置：光盘\MR\06\302	高级 趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 类库在同一图像中生成多个饼形图，对公司 2009 年上半年编程词典销量进行分析，其运行结果如图 6.34 所示。

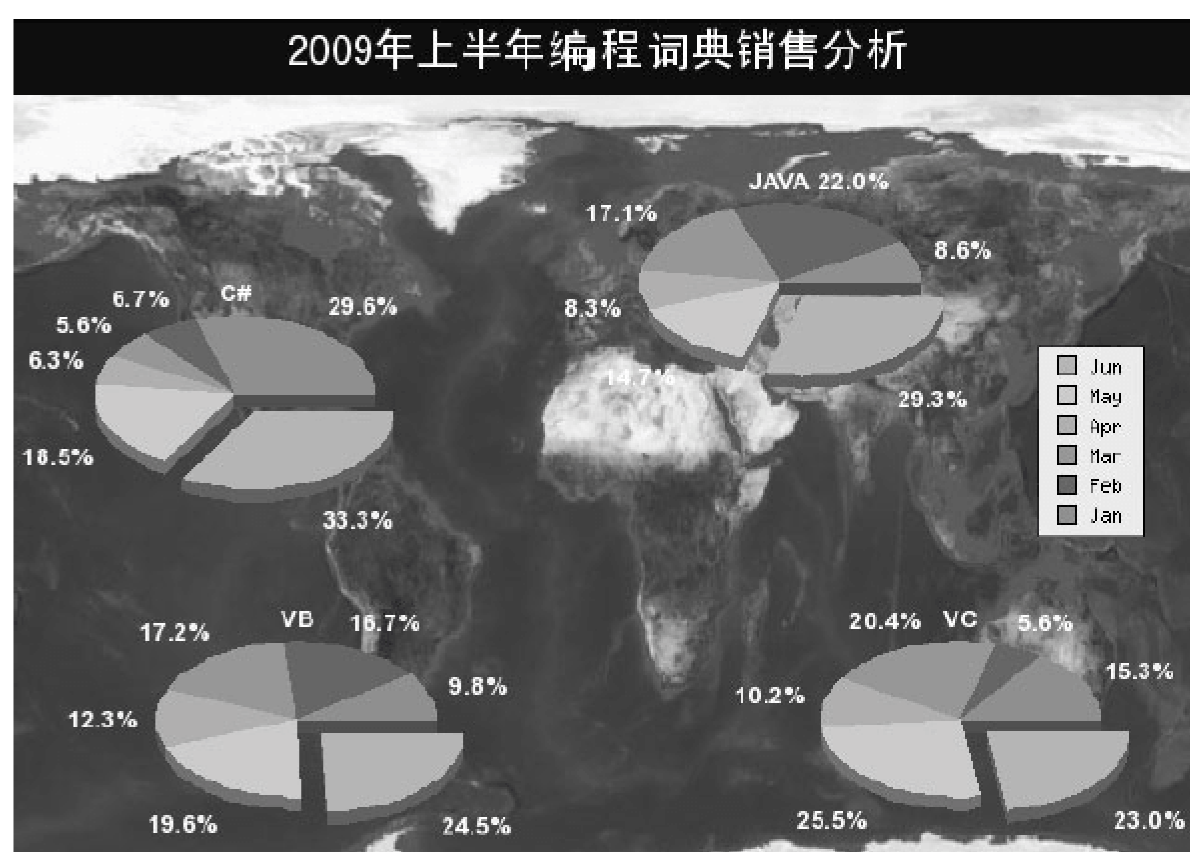


图 6.34　2009 年上半年编程词典销量分析

关键技术

在本实例中，通过 Jpgraph 类库创建多饼形图，其实现的关键是数组和 for 循环语句的运用。饼形图创建的原理与上面介绍的是相同的，只是这里根据传递的数组数据，对饼形图的创建方法执行了循环操作，进而完成多饼形图的创建。

其中关键的内容就是输出的数据、饼形图的坐标定义和饼形图标题的设置，根据这 3 个数组数据，然后执行 for 循环，完成饼形图的创建，其关键代码如下：

```
$data = array(                                          //定义输出的数据
    array(80,18,15,17,50,90),
    array(35,90,70,34,60,120),
    array(40,68,70,50,80,100),
    array(60,22,80,40,100,90));
$piepos = array(0.2,0.4,0.65,0.28,0.25,0.75,0.8,0.75);  //定义饼形图的坐标
$titles = array('C#','JAVA','VB','VC');                 //定义标题
$n = count($piepos)/2;                                  //统计饼形图的个数
```

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）根据定义的坐标值数组，计算出饼形图的个数，应用 Jpgraph 类库中的方法，通过 for 循环创建多个

饼形图，其代码如下：

```
$graph = new PieGraph(650,500,'auto');                          //创建图像
$graph->SetMargin(1,1,40,1);                                    //定义图像的边距
$graph->SetMarginColor('navy');                                 //定义边距的颜色
$graph->SetShadow(false);
$graph->SetBackgroundImage('worldmap1.jpg',BGIMG_FILLPLOT);     //添加背景图像
$graph->title->Set(iconv("utf-8","gb2312",'2009 年上半年编程词典销售分析'));
$graph->title->SetFont(FF_SIMSUN, FS_BOLD,18);
$graph->title->SetColor('white');
$p = array();                                                   //定义空数组
for( $i=0; $i < $n; ++$i ) {                                    //根据饼形图的个数
    $d = "data$i";
    $p[] = new PiePlot3D($data[$i]);                            //创建 3D 饼形图
}
for( $i=0; $i < $n; ++$i ) {
    $p[$i]->SetCenter($piepos[2*$i],$piepos[2*$i+1]);           //定义饼形图的位置
}
for( $i=0; $i < $n; ++$i ) {                                    //循环输出饼形图的标题
    $p[$i]->title->Set($titles[$i]);                            //输出标题
    $p[$i]->title->SetColor('white');                           //定义颜色
    $p[$i]->title->SetFont(FF_ARIAL,FS_BOLD,10);                //定义字体
}
for( $i=0; $i < $n; ++$i ) {                                    //循环读取饼形图的数据
    $p[$i]->value->SetFont(FF_ARIAL,FS_BOLD);                   //定义字体
    $p[$i]->value->SetColor('white');                           //定义颜色
}
for( $i=0; $i < $n; ++$i ) {                                    //设置数据的输出阴影
    $p[$i]->value->Show();
}
for( $i=0; $i < $n; ++$i ) {                                    //定义输出数据的格式
    $p[$i]->value->SetFormat("%01.1f%%");                       //数据的格式化
}
for( $i=0; $i < $n; ++$i ) {                                    //定义图像大小
    $p[$i]->SetSize(0.15);
}
for( $i=0; $i < $n; ++$i ) {                                    //定义分割的饼形图块
    $p[$i]->SetEdge(false);
    $p[$i]->ExplodeSlice(5);
}
$p[0]->SetLegends(array("Jan","Feb","Mar","Apr","May","Jun"));  //定义注释内容
$graph->legend->Pos(0.05,0.35);                                 //定义注释的坐标
$graph->legend->SetShadow(false);
for( $i=0; $i < $n; ++$i ) {                                    //循环执行饼形图数据的添加操作
    $graph->Add($p[$i]);
}
$graph->Stroke();                                               //生成图像
```

秘笈心法

心法领悟 302：同一图像中多饼形图的创建。

在同一图像中创建多个饼形图的关键是如何完成数据的循环读取，只要能够准确地定位每个饼形图在图像中的位置，并且执行正确的循环操作，那么多饼形图的创建就不是问题，至于创建的方法与单个饼形图是相同的。

实例 303	环饼形图分析 2009 年图书销量 光盘位置：光盘\MR\06\303	高级 趣味指数：★★★★☆

实例说明

在本实例中，运用 Jpgraph 类库生成环饼形图，对公司 2009 年图书销量进行分析，其运行结果如图 6.35 所示。

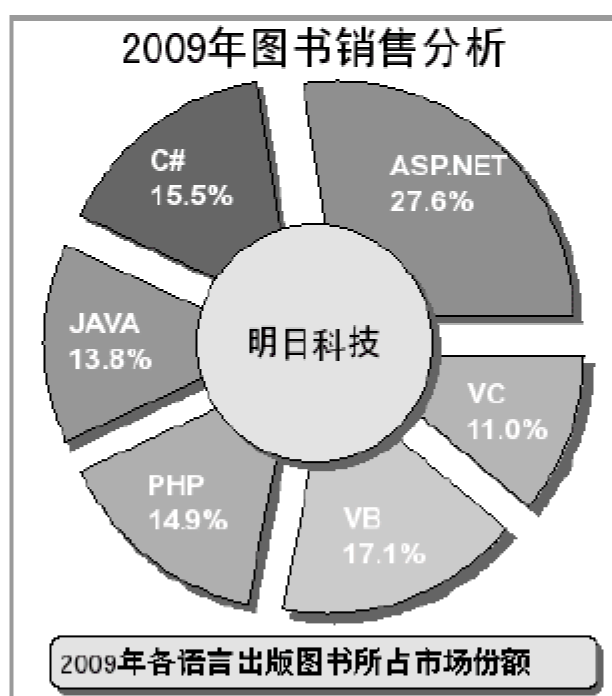


图 6.35　2009 年图书销量分析

关键技术

在本实例中，通过 Jpgraph 类库创建环饼形图，对 2009 年各部门图书销量进行分析，并且通过创建的文本输出说明信息，其应用的关键技术如下：

（1）通过 PiePlotC 类创建环饼形图图像，该类继承 PiePlot 类。

（2）通过环饼形图中的 midtitle 对象调用 Set()方法输出环饼形图的中心标题，调用 SetFont()方法设置中心标题的字体。

（3）通过 Text 类在图像中定义文本，其参数为向文本中添加的数据。

设计过程

（1）创建 index.php 文件，设置网页的编码格式，并通过 include()语句导入所需的存储在章文件夹 06 下的 Jpgraph 文件。

（2）应用 Jpgraph 类库中的方法创建一个环饼形图，对 2009 年图书销量进行分析，并通过文本输出说明信息，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include ("../src/jpgraph.php");
include ("../src/jpgraph_pie.php");
$data = array(50,28,25,27,31,20);
$graph = new PieGraph(400,400);                                         //创建图像
$graph->title->Set(iconv("utf-8","gb2312",'2009 年图书销售分析'));        //设置标题文字
$graph->title->SetFont(FF_SIMSUN, FS_BOLD,18);                          //设置字体
$graph->title->SetMargin(8);                                            //设置边距
$p1 = new PiePlotC($data);                                              //创建图像
$p1->SetSize(0.32);                                                     //设置图像大小
$p1->SetCenter(0.5,0.45);                                               //定义图像存储位置
$p1->value->SetFont(FF_ARIAL,FS_BOLD,12);                               //设置数据的字体
$p1->value->SetColor('white');                                          //设置字体颜色
$p1->midtitle->Set(iconv("utf-8","gb2312","明日科技"));                  //输出中心内容
$p1->midtitle->SetFont(FF_SIMSUN, FS_BOLD,14);                          //设置字体
$p1->SetMidColor('yellow');                                             //设置中心图像颜色
$lbl = array("ASP.NET\n%.1f%%","C#\n%.1f%%","JAVA\n%.1f%%",
        "PHP\n%.1f%%","VB\n%.1f%%","VC\n%.1f%%");                        //定义数据内容
$p1->SetLabels($lbl);
$p1->SetShadow();                                                       //生成阴影
$p1->ExplodeAll(20);                                                    //定义图像大小
$txt = new Text(iconv("utf-8","gb2312","2009 年各语言出版图书所占市场份额"));  //创建文本，并添加数据
$txt->SetFont(FF_SIMSUN, FS_BOLD,12);
$txt->Pos(0.5,0.97,'center','bottom');                                  //设置存储位置
$txt->SetBox('yellow','black');                                         //设置颜色
$txt->SetShadow();                                                      //设置图像阴影
```

```
$graph->AddText($txt);                                          //添加数据
$graph->Add($p1);                                               //向图像中添加数据
$graph->StrokeCSIM();                                           //生成图像
?>
```

秘笈心法

心法领悟 303：Jpgraph 类库的应用总结。

在 6.3 节的实例中，笔者对 Jpgraph 类库中常用的图像创建方法进行了讲解。由于笔者的水平有限，在讲解过程中难免有疏漏之处，还请读者见谅。

最后，教大家一个学习和运用 Jpgraph 类库的好方法。读者在下载 Jpgraph 类库时，在下载的文件中包含一个 docportal 文件夹，在这个文件夹中存储了 Jpgraph 类库中方法、对象和类的详细说明，读者如果有不明之处可以在这个参考手册中寻找答案。另外，在下载的 src 文件夹中，还包含了一个 Examples 文件夹，在这个文件夹中存储了很多 Jpgraph 类库创建图像的实例，读者可以根据自己的需求进行选择。

6.4 GD2 函数库

GD2 函数库在图像图形绘制方面功能非常强大，开发人员既可以在已有图片的基础上进行绘制，也可以在没有任何素材的基础上绘制。

实例 304 绘制基本的几何图形 中级

光盘位置：光盘\MR\06\304 趣味指数：★★★★☆

实例说明

应用 GD2 函数可以绘制的图形有多种，最基本的图形包括线条、圆、方形等，无论开发人员绘制多么复杂的图形都是在这些最基本的图形基础上进行深化的，只有掌握了最基本图形的绘制方法，才能够绘制出各种具有独特风格的图形。在本实例中，将讲解这些基本几何图形的绘制方法，运行结果如图 6.36 所示。

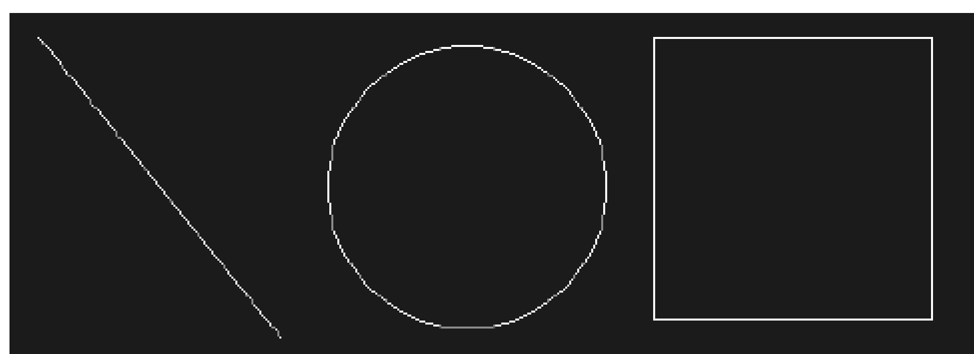


图 6.36 绘制基本几何图形

关键技术

在 GD2 函数库中分别应用 imageline()函数、imagearc()函数和 imagerectangle()函数绘制直线、圆形和方形。

imageline()函数，用于绘制直线，其语法如下：

```
bool imageline ( resource image, int x1, int y1, int x2, int y2, int color )
```

该函数用颜色 color 在图像 image 中从坐标（x1，y1）到（x2，y2）（图像左上角坐标为（0, 0））绘制一条线段。

imagearc()函数，用于绘制圆形，其语法如下：

```
bool imagearc ( resource image, int cx, int cy, int w, int h, int s, int e, int color )
```

imagearc()函数的参数说明如表 6.3 所示。

表 6.3　imagearc()函数的参数说明

参　数	说　明
image	表示图像的 handle
cx、cy	原点坐标（0,0）为图片的左上角，参数 cx、cy 为椭圆圆心坐标
w、h	参数 w、h 为水平轴长和垂直轴长
s、e	分别为起始角与结束角
color	弧线的颜色

imagerectangle()函数，用于绘制方形，其语法如下：

```
bool imagerectangle ( resource image, int x1, int y1, int x2, int y2, int col )
```

该函数用颜色 col 在图像 image 中绘制一个方形，其左上角坐标为（x1, y1），右下角坐标为（x2, y2）。图像的左上角坐标为（0, 0）。

设计过程

应用 GD2 函数最基本的图形绘制函数，分别绘制直线、正圆和正方形，并用白色作为线条基色，代码如下：

```
<?php
header("Content-type: image/png");                    //将图像输出到浏览器
$img = imagecreate(560, 200);                         //创建一个 560×200 的图像
$bg = imagecolorallocate($img, 0, 0, 255);            //设置图像背景色
$white = imagecolorallocate($img, 255, 255, 255);     //设置绘制图像的颜色为白色
imageline($img, 20, 20, 150, 180, $white);            //绘制一条直线
imagearc($img, 250, 100, 150, 150, 0, 360, $white);   //绘制一个圆
imagerectangle($img, 350, 20, 500, 170, $white);      //绘制一个正方形
imagepng($img);                                       //以.png 格式输出图形
imagedestroy($img);                                   //释放资源
?>
```

秘笈心法

心法领悟 304：在 PHP 中如何激活和屏蔽 GD2 函数库。

GD 库是在 PHP 5 中默认安装的，但是要想激活 GD 库，必须修改 php.ini 文件。将该文件中的;extension=php_gd2.dll 选项前的分号“;”去掉，保存文件并重新启动 Apache 服务器即可生效。如果要屏蔽对 GD2 函数库的支持，则在 extension=php_gd2.dll 选项前加上分号即可。

实例 305	GD2 函数填充几何图形 光盘位置：光盘\MR\06\305	中级 趣味指数：★★★★☆

实例说明

使用 GD2 函数不仅可以绘制线条图形，还可以绘制填充图形，如填充圆形、填充矩形等，在本实例中，将介绍圆形和矩形的填充，其运行结果如图 6.37 所示。



图 6.37　填充几何图形

关键技术

在 GD2 函数库中，应用 imagefill()函数实现图像的填充操作，其语法如下：

```
bool imagefill ( resource image, int x, int y, int color)
```

该函数在图像 image 的坐标（x，y）（图像左上角坐标为（0, 0））处用颜色 color 执行区域填充（即与（x, y）点颜色相同且相邻的点都会被填充）。

应用 imagefilledarc()函数绘制一个椭圆弧线，其语法如下：

```
bool imagefilledarc ( resource image, int cx, int cy, int w, int h, int s, int e, int color, int style )
```

该函数在 image 所代表的图像中以（cx，cy）（图像左上角坐标为（0, 0））为坐标点绘制一椭圆弧。如果成功返回 TRUE，否则返回 FALSE。参数 w 和 h 分别指定椭圆的宽和高；s 和 e 以角度指定起始和结束点；style 可以是下列值按位或（OR）后的值：

☑ IMG_ARC_PIE。
☑ IMG_ARC_CHORD。
☑ IMG_ARC_NOFILL。
☑ IMG_ARC_EDGED。

IMG_ARC_PIE 和 IMG_ARC_CHORD 是互斥的。IMG_ARC_CHORD 只是用直线连接起始和结束点，而 IMG_ARC_PIE 则产生圆形边界（如果两个都用，IMG_ARC_CHORD 生效）；IMG_ARC_NOFILL 指明弧或弦只有轮廓，不填充；IMG_ARC_EDGED 指明用直线将起始和结束点与中心点相连，和 IMG_ARC_NOFILL 一起使用是绘制饼状图轮廓的好方法（不用填充）。

应用 Imagefilledellipse()函数绘制一个椭圆，其语法如下：

```
bool imagefilledellipse ( resource image, int cx, int cy, int w, int h, int color )
```

该函数在 image 所代表的图像中以（cx，cy）（图像左上角坐标为（0, 0））为中心绘制一个椭圆。参数 w 和 h 分别指定椭圆的宽和高，color 指定填充颜色。如果成功返回 TRUE，否则返回 FALSE。

设计过程

本实例应用 GD2 函数绘制填充椭圆形和填充正方形，其代码如下：

```
<?php
header("Content-type: image/png");                    //将图像输出到浏览器
$img = imagecreate(400, 200);                         //创建一个 400×180 的图像
$bg = imagecolorallocate($img, 0, 0, 255);
$white = imagecolorallocate($img, 255, 0, 255);
imagefilledellipse($img, 100, 100, 150, 150, $white);      //绘制圆形
imagefilledrectangle($img, 200, 50, 300, 150, $white);     //绘制正方形
imagepng($img);
imagedestroy($img);
?>
```

秘笈心法

心法领悟 305：通过 GD2 函数库创建图像必不可少的函数。

在通过 GD2 函数库创建图像时，imagedestroy()函数是必不可少的，可以通过它销毁指定的图像，释放与 image 关联的内存。如果不应用该函数，与 image 关联的数据将会一直存储在内存中。

实例 306	GD2 函数输出英文字符串 光盘位置：光盘\MR\06\306	初级 趣味指数：★★✮

实例说明

在本实例中，介绍如何在图像中输出字符串（注意这里介绍的是如何输出英文字符串），其运行效果如

图 6.38 所示。

Do you still love php tomorrow ?

图 6.38　输出英文字符串

关键技术

GD2 函数库中应用 imageString()函数在画布上输出字符串，其语法如下：

```
bool imagestring ( resource image, int font, int x, int y, string s, int col )
```

该函数用颜色 col 将字符串 s 绘制到 image 所代表的图像的（x，y）处（这是字符串的左上角坐标，整幅图像的左上角坐标为（0，0））。如果 font 是 1、2、3、4 或 5，则使用内置字体。

设计过程

本实例应用 imageString()函数在画布上水平地绘制一行英文字符串，其代码如下：

```
<?php
header("Content-type: image/png");
$im = imagecreate(600, 50);
$bg = imagecolorallocate($im, 255, 255, 255);
$red = imagecolorallocate($im, 255, 0, 0);
imagestring($im, 5, 0, 0, "Do you still love php tomorrow ?", $red);         //在指定的坐标处，水平地绘制一行红色的字符串
imagepng($im);                                                                //输出图像
imagedestroy($img);                                                           //销毁图像
?>
```

秘笈心法

心法领悟 306：如果要输出中文字符串应该怎么办？

在 GD2 函数库中，想要输出中文字符串，必须将中文字符串的编码格式转换成 utf-8，然后应用 imageTTFText()函数完成中文字符串的输出。如果直接使用 imagestring()函数，将输出乱码。

实例 307	GD2 函数在照片上添加文字 光盘位置：光盘\MR\06\307	中级 趣味指数：★★★★

实例说明

PHP 中的 GD 库支持中文，但必须要以 utf-8 格式的参数来进行传递，如果使用 imageString()函数直接绘制中文字符串就会显示乱码。这是因为 GD2 对中文只能接收 utf-8 编码格式，并且默认使用了英文的字体，所以要输出中文字符串，必须对其进行转码，并设置中文字符使用的字体。否则，输出的只能是乱码。在本实例中，实现 GD2 函数输出中文字符串，并且通过 GD2 函数将中文字符串在照片上输出，其运行结果如图 6.39 所示。



图 6.39　在照片上输出中文字符串

关键技术

在 PHP 中，在图像中添加中文字符串应用的是 imagettftext ()函数，其语法如下：

```
array imagettftext ( resource image, float size, float angle, int x, int y, int color, string fontfile, string text )
```

imagettftext ()函数的参数说明如表 6.4 所示。

表 6.4　imagettftext ()函数的参数说明

参　数	说　明
image	图像资源
size	字体大小。根据 GD 版本不同，应该以像素大小（GD1）或点大小（GD2）指定
angle	字体的角度，顺时针计算，0° 为水平，也就是 3 点钟的方向（由左到右），90° 则为由下到上的文字
x	文字的 X 坐标值。设定第一个字符的基本点
y	文字的 Y 坐标值。设定字体基线的位置，不是字符的最底端
color	文字的颜色
fontfile	字体的文件名称，也可以是远端的文件
text	字符串内容

在 PHP 中，通过 GD2 函数对 JPG 格式的照片进行操作应用 imagecreatefromjpeg()函数，根据 JPEG 文件或 URL 新建一图像，其语法如下：

```
resource imagecreatefromjpeg ( string filename )
```

参数 filename 可以是本地文件，也可以是网络的 URL 地址。返回值为 JPEG 的文件代码，可供其他函数使用。该函数在失败时返回一个空字符串，并且输出一条错误信息。

设计过程

本实例应用 imagecreatefromjpeg()和 imagettftext ()函数在照片上添加中文字符串，其代码如下：

```
<?php
header("content-type:image/jpeg");                    //定义输出为图像类型
$im=imagecreatefromjpeg("images/P1020494.JPG");       //载入照片
$textcolor=imagecolorallocate($im,25,25,136);         //设置字体颜色为蓝色，值为 RGB 颜色值
$font="Font/mzd.ttf";                                 //定义字体
$to="落霞与孤鹜齐飞";
imagettftext($im,80,0,200,200,$textcolor,$font,$to);  //写 TTF 文字到图中
$to="秋水共长天一色";
imagettftext($im,80,0,300,400,$textcolor,$font,$to);  //写 TTF 文字到图中
imagejpeg($im);                                       //建立 JPEG 图形
imagedestroy($im);                                    //结束图形，释放内存空间
?>
```

秘笈心法

心法领悟 307：必须牢记一点，imagettftext()函数只支持 utf-8 编码。

因为本实例中页面使用的是 utf-8 编码格式，所以在定义向图片中添加的中文字符串时可以直接使用，不需要编码格式的转换。但是，如果页面本身使用的是 gb2312 编码格式，那么就必须应用 iconv()函数对向图片中添加的中文字符串进行编码格式的转换，由 gb2312 编码转换为 utf-8 编码。

实例 308　GD2 函数为图片添加文字水印　高级

光盘位置：光盘\MR\06\308　趣味指数：★★★★☆

实例说明

图片是 Web 页面最为重要的组成元素之一，新闻网站、图片资料网站等备受网民关注的网站每天都会上传大量的图片。如果直接将图片上传到页面中，很可能被浏览者保存使用，这样站点的版权就不能很好地得到保证，如果在上传图片过程中，动态地为图片添加水印效果，这样不仅可以对保证站点版权起到一定作用，如果设计合理还能有助于网站的推广。在本实例中，将讲解如何在上传图片的过程中为图片添加水印文字，其运行结果如图 6.40 所示。



图 6.40　为上传图片添加文字水印

关键技术

为上传图片添加水印的重点是图像处理函数的运用，至于文件的上传已经在前面的实例中进行了详细的讲解，这里就不再赘述。本实例运用的关键技术如下：

（1）创建图像函数

创建图像函数的运用是完成本实例的第一步，因为需要根据上传图片的后缀（格式）运用不同的函数来创建新图像，这里定义 3 种格式的图片（gif、jpg 和 png），所以要应用到 3 个创建图像函数，分别为：

☑ imagecreatefromgif()函数，根据 GIF 文件或者 URL 新建一图像，返回图像标识符。

☑ imagecreatefromjpeg()函数，根据 JPG 文件或者 URL 新建一图像，返回图像标识符。

☑ imagecreatefrompng()函数，根据 PNG 文件或者 URL 新建一图像，返回图像标识符。

（2）操作图像函数

新图像创建成功后将返回一个图像标识符，操作图像函数根据标识符对图像进行操作，这里应用 2 个函数：

imagecolorallocate()函数：定义水印文字的颜色。

imagettftext()函数：在返回的图像中输入中文字符串。

（3）输出图像函数

与创建图像函数对应，同样有 3 个输出图像函数：

☑ imagegif()函数，以 GIF 格式将图像输出到浏览器或文件。

☑ imagejpeg()函数，以 JPG 格式将图像输出到浏览器或文件。

☑ imagepng()函数，以 PNG 格式将图像输出到浏览器或文件。

（4）销毁图像函数

在完成图像的创建和输出之后，有必要销毁图像，释放与 image 相关的内存，其应用的是 imagedestroy()函数，语法如下：

```
bool imagedestroy ( resource image )
```

该函数释放与 image 关联的内存。参数 image 是由图像创建函数返回的图像标识符。

在本实例中，要对上述函数进行整合应用，创建 AddWaterPress 类，定义 getExtendsName()方法，获取上传

图片的文件后缀；定义 getImageRes()方法，根据上传文件的后缀创建新图像；定义 outputImage()方法，输出图像；定义 add()方法，向指定的图片中添加文字水印。

设计过程

（1）创建 index.php 文件。首先，创建一个表单，完成上传图片的提交操作。然后，在本页中输出存储在服务器指定文件夹下的图片。最后，获取表单中提交的图片数据，应用 move_uploaded_file()函数完成图片的上传操作，并且实例化存储在 AddWaterPress.php 文件中的 AddWaterPress 类，调用 add()方法为指定的图片添加文字水印，其关键代码如下：

```
if(move_uploaded_file($_FILES["file"]["tmp_name"], $saveDir)){                //执行文件上传操作
require_once 'AddWaterPress.php';                                              //包含添加水印操作的文件
    $addWaterPress = new AddWaterPress();                                      //类的实例化
    $addWaterPress->add($saveDir, "吉林省明日科技");                           //执行添加方法，传递参数，指定水印文字
      echo "<script>alert('图片添加成功');</script>";
}
```

（2）创建 AddWaterPress.php 文件，编写 AddWaterPress 类的内容，其关键代码如下：

```
<?php
class AddWaterPress{                                                           //定义类文件
//省略了部分代码
 function add($imageUrl, $watherImageUrl){                                     //定义添加方法
        $img = @$this->getImageRes($this->getExtendsName($imageUrl), $imageUrl);  //获取被操作的图像标识
        $textcolor=imagecolorallocate($img,190,1,23);                          //设置字体颜色为蓝色，值为 RGB 颜色值
        $font="Font/FZHCJW.TTF";                                               //定义字体
        imagettftext($img,15,56,20,130,$textcolor,$font,$watherImageUrl);      //写 TTF 文字到图中
        //根据图像标识符、后缀和路径，执行 outputImage 方法，输出图像
       $this->outputImage($img, $this->getExtendsName($imageUrl), $imageUrl);
      imagedestroy($img);                                                      //销毁图像
}
}
```

秘笈心法

心法领悟 308：添加英文字符串作为水印标记。

本实例中完成的是中文字符串水印标记的添加操作，如果要添加英文字符串格式的水印标记，可以直接使用 imageString()函数来完成，其方法与在图像中输出英文字符串是相同的。

实例 309	GD2 函数为图片添加图像水印 光盘位置：光盘\MR\06\309	初级 趣味指数：★★★

实例说明

实例 308 介绍的是如何为照片添加文字水印，在本实例中将介绍如何为上传图片添加一个图像水印，其运行结果如图 6.41 所示。



图 6.41　为图片添加图像水印

关键技术

使用图片作为水印的前提是这个图片的背景必须是透明的，否则输出的效果很不理想。添加图片水印的关键是 getimagesize()和 imagecopy()函数。应用 getimagesize()函数获取上传图片和水印图片的大小，通过 imagecopy()函数完成图片水印的添加。

imagecopy()函数，将图像复制到指定的另外一个图像中，其语法如下：

```
bool imagecopy ( resource dst_im, resource src_im, int dst_x, int dst_y, int src_x, int src_y, int src_w, int src_h )
```

将图像 src_im 中坐标从（src_x，src_y）开始，宽度为 src_w，高度为 src_h 的一部分复制到图像 dst_im 中坐标为（dst_x，dst_y）的位置上。

在本实例中，同样是整合图像的创建、操作、输出和销毁函数，创建 AddWaterPress 类，定义 getExtendsName()方法，获取上传图片的文件后缀；定义 getImageRes()方法，根据上传文件的后缀创建新图像；定义 outputImage()方法，输出图像；定义 add()方法，向指定的图片中添加图像水印。

设计过程

本实例的实现步骤与实例 308 相同，唯一的区别是创建 AddWaterPress.php 文件，编写 AddWaterPress 类的 add()方法时，应用的是 getimagesize()和 imagecopy()函数完成图像水印的添加操作。Add()方法的代码如下：

```
function add($imageUrl, $watherImageUrl, $x, $y){                               //定义添加方法
        $img = @$this->getImageRes($this->getExtendsName($imageUrl), $imageUrl);  //获取被添加的图像标识
        $img1 = @$this->getImageRes($this->getExtendsName($watherImageUrl), $watherImageUrl);
        $size = getimagesize($imageUrl);                                          //获取图像大小
       $size1 = getimagesize($watherImageUrl);                                    //获取水印图片的大小
        if($x==null && $y==null){                                                 //判断参数是否为空
              $x1 = ($size[0]-$size1[0])/2;                                       //根据图像大小数组中返回的值，计算图像的横坐标
                $y1 = ($size[1]-$size1[1])/2;                                     //根据图像大小数组中返回的值，计算图像的纵坐标
        }else{
              $x1 = $x;                                                           //如果不为空，则直接使用坐标数据
                $y1 = $y;                                                         //如果不为空，则直接使用坐标数据
        }
       imagecopy($img, $img1, $x1, $y1, 0, 0, $size1[0], $size1[1]);              //将 img1 的一部分复制到 img 的指定位置
        $this->outputImage($img, $this->getExtendsName($imageUrl), $imageUrl);    //根据图像标识符、后缀和路径，执行 outputImage()方法，
                                                                                   输出图像
        imagedestroy($img1);                                                      //销毁图像
       imagedestroy($img);                                                        //销毁图像
}
```

秘笈心法

心法领悟 309：使用背景透明的图片作为水印图片效果更好。

在将图片作为水印图像时，如果指定的图片背景不是透明的，那么添加的水印图像将与原始图像格格不入，而如果水印图片的背景是透明的，那么添加的水印标记就可以很好地与原始图像融入一体。

实例 310	GD2 函数生成图形验证码 光盘位置：光盘\MR\06\310	高级 趣味指数：★★★★☆

实例说明

验证码技术的应用，是为了提高站点的安全性，避免因网页运行速度慢而造成数据的重复提交。本实例中，将通过 JavaScript 脚本和 GD2 函数开发一个无刷新验证码，其运行结果如图 6.42 所示。

图 6.42　GD2 函数生成图形验证码

关键技术

应用 JavaScript 脚本和 GD2 函数开发一个无刷新验证码，涉及的技术包括随机数的生成、随机数如何传递给图像生成页、数字图像的生成以及验证码的二次生成。这里重点介绍验证码图像的生成。

（1）创建 ValidatorCode.php 页面，应用 GD2 函数库中的函数，根据超链接传递的参数值生成验证码图像。应用的函数主要有以下几点：

☑ imagecreate()函数，创建验证码图像的画布；
☑ imagecolorallocate()函数，定义画布的填充颜色；
☑ imagefill()函数，根据 imagecolorallocate()函数定义的颜色，完成图像颜色的填充操作；
☑ imagestring()函数，将从超链接中获取的验证码值写入到图像中，并且应用 imagecolorallocate()函数随机定义值输出的颜色；
☑ imagepng()函数，生成 PNG 格式的图像；
☑ imagedestroy()函数，销毁图像，释放内存。

（2）验证码的二次生成。其实现原理类似于 do...while 循环语句，无论条件是否满足，默认都会执行一次（打开网页后，直接生成一个验证码）。然而，要想使其继续执行，就必须满足指定的条件（重新生成验证码，就必须调用指定的方法）。依据这个原理，在 JavaScript 脚本中定义如下方法：

```
function reCode(){                                                    //定义方法，重新生成验证码
 var num1=Math.round(Math.random()*10000000);                         //生成随机数
 var num=num1.toString().substr(0,4);                                 //截取随机数
 document.codeimg.src="ValidatorCode.php?code="+num;                  //将截取值传递到图像处理页中
 form1.defValidatorCode.value=num;                                    //将截取值赋给表单中的隐藏域
}
```

在上述方法中，重复执行 JavaScript 脚本中默认的操作，即重新生成一个验证码。

设计过程

（1）创建 index.php 页面，定义 form 表单，设计用户注册页面的效果。

（2）编写 JavaScript 脚本，生成验证码，并定义 reCode()方法，用于验证码的二次生成，其关键代码如下：

```
<script language="javascript">
var num1=Math.round(Math.random()*10000000);                          //生成随机数
var num=num1.toString().substr(0,4);                                  //截取随机数的前 4 个字符
document.write("<img name=codeimg src='ValidatorCode.php?code="+num+"'>");   //将值传递到图像页
form1.defValidatorCode.value=num;                                     //将截取值赋给表单中的隐藏域
</script>
```

（3）创建超链接，调用 JavaScript 脚本中的 reCode()方法，实现验证码的二次生成，其代码如下：

```
<a href="javascript:reCode()" class="a1">看不清</a>
```

（4）创建 ValidatorCode.php 页面，根据超链接中传递的随机数，应用 GD2 函数生成验证码的数字图像，其关键代码如下：

```
<?php
header('content-type:image/png');                                     //定义标题.png 格式图像
$im = imagecreate(65, 25);                                            //定义画布
imagefill($im, 0, 0, imagecolorallocate($im, 200, 200, 200));         //区域填充
$validatorCode = $_GET['code'];                                       //获取提交的值
imagestring($im, rand(3, 5), 10, 3, substr($validatorCode, 0, 1), imagecolorallocate($im, 0, rand(0, 255), rand(0, 255)));
imagestring($im, rand(3, 5), 25, 6, substr($validatorCode, 1, 1), imagecolorallocate($im, rand(0, 255), 0, rand(0, 255)));
imagestring($im, rand(3, 5), 36, 9, substr($validatorCode, 2, 1), imagecolorallocate($im, rand(0, 255), rand(0, 255), 0));
imagestring($im, rand(3, 5), 48, 12, substr($validatorCode, 3, 1), imagecolorallocate($im, 0, rand(0, 255), rand(0, 255)));
imagepng($im);                                                        //生成 PNG 图像
imagedestroy();                                                       //销毁图像
```

（5）在 js 文件夹下，创建一个 js 脚本文件，定义 chkinput()方法，对表单中提交的注册信息进行验证，包括验证码是否正确。其完整代码请参考本书光盘。

秘笈心法

心法领悟 310：生成带有干扰线的数字图形验证码。



图 6.43　增加干扰背景后的验证码图像

在本实例中，应用 GD2 函数生成的是清晰的数字图形验证码，还可以在此基础上为验证码增加干扰背景，使其看上去模糊一些，如图 6.43 所示。

其具体方法如下：在 ValidatorCode.php 文件中，通过 for 循环语句，应用 imagesetpixel()函数在画布的背景上绘制一些单一元素，代码如下：

```
for ($i = 0; $i < 200; $i ++) {                                          //填充干扰背景
    imagesetpixel($im, rand() % 70, rand() % 30, imagecolorallocate($im, rand(0, 255), rand(0, 255), rand(0, 255)));
}
```

实例 311	GD2 函数折线图分析网站月访问量走势 光盘位置：光盘\MR\06\311	高级 趣味指数：★★★★☆

实例说明

在本实例中，运用 GD2 函数自行编写一个绘制折线图的方法，对 2009 年公司网站的月访问量进行分析。本实例中的方法完全由笔者自行编写，不借助于任何图像的操作类库，其运行结果如图 6.44 所示。

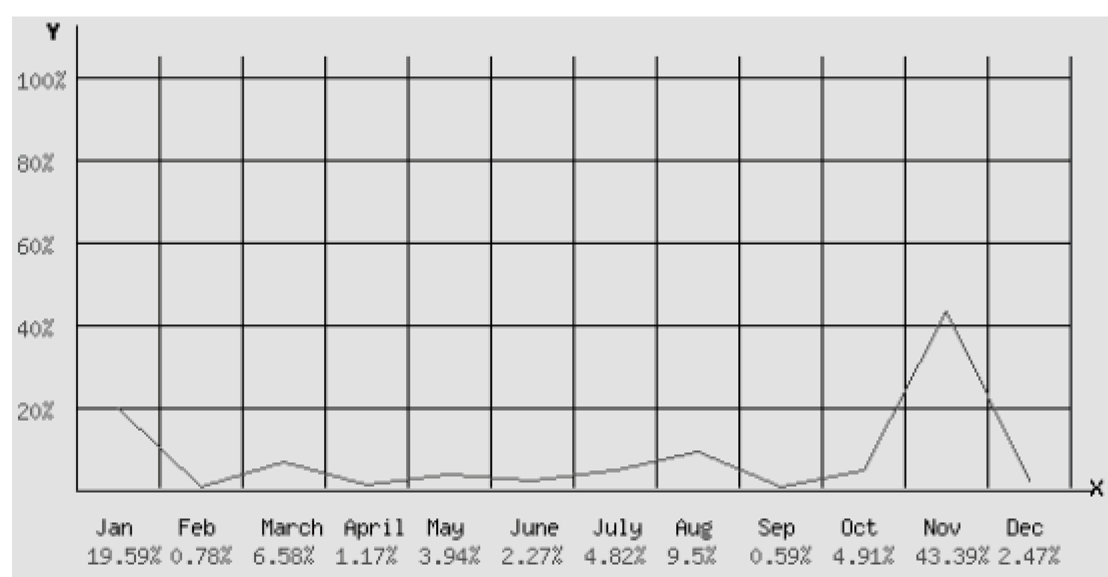


图 6.44　GD2 折线图分析网站月访问量走势

关键技术

本实例中，通过 GD2 函数绘制网站访问量走势分析图，其中涉及的关键技术如下：

（1）应用 imageline()函数绘制线条。

imageline()函数用指定颜色在图像中两点之间绘制一条线段，语法如下：

```
bool imageline ( resource image, int x1, int y1, int x2, int y2, int color )
```

参数说明：

image：图像标识。

x1：起始点的横坐标。

y1：起始点的纵坐标。

x2：结束点的横坐标。

y2：结束点的纵坐标。

color：颜色标识。

（2）应用百分比来显示数据信息，并通过 round()函数将数字小数位进行四舍五入操作，round()函数的语法如下：

```
double round(double val)
```

参数 var：进行四舍五入操作的小数。

设计过程

（1）编写 index.php 文件，创建表单，添加 12 个表单元素，提交网站的月访问量数据。

（2）创建 img.php 文件，获取表单中提交的网站月访问量数据，并对数据统计分析。通过 GD2 函数创建一个折线图输出统计分析的结果，其关键代码如下：

```
<?php
$data = array ($_POST ["T1"], $_POST ["T2"], $_POST ["T3"], $_POST ["T4"], $_POST ["T5"], $_POST ["T6"], $_POST ["T7"], $_POST ["T8"],
$_POST ["T9"], $_POST ["T10"], $_POST ["T11"], $_POST ["T12"] );
$month = array ("Jan", "Feb", "March", "April", "May", "June", "July", "Aug", "Sep", "Oct", "Nov", "Dec" );
$max = 0;
for($i = 0; $i < 12; $i ++) {
        $max = $max + $data [$i];                                          //所有网站访问量的累加和
}
$im = imagecreate ( 550, 300 );                                            //创建画布
$green = imagecolorallocate ( $im, 214, 235, 214 );                        //设置颜色值
$black = imagecolorallocate ( $im, 0, 0, 0 );
$red = imagecolorallocate ( $im, 255, 0, 0 );
$blue = imagecolorallocate ( $im, 0, 0, 255 );
imageline ( $im, 30, 230, 520, 230, $blue );                               //设置 X 轴横坐标
imageline ( $im, 30, 5, 30, 230, $blue );                                  //设置 Y 轴纵坐标
imagestring ( $im, 3, 520, 222, "X", $black );                             //输出字符 X
imagestring ( $im, 3, 16, 1, "Y", $black );                                //输出字符 Y
$l = 190;
$k1 = 30;
$k2 = 510;
for($j = 0; $j < 12; $j ++) {
        imageline ( $im, $k1, $l, $k2, $l, $black );                       //设置 X 轴网格线横坐标
        $l = $l - 40;
}
$f = 70;
$z1 = 20;
$z2 = 228;
for($j = 0; $j < 12; $j ++) {
        imageline ( $im, $f, $z1, $f, $z2, $black );                       //设置 Y 轴网格线纵坐标
        $f = $f + 40;
}
$l = 185;
for($j = 1; $j < 6; $j ++) {
        imagestring ( $im, 2, 2, $l, 20 * $j . "%", $red );
        $l = $l - 40;
}
$x = 20;
$y = 230;
for($i = 1; $i < 12; $i ++) {
        $y_lt = $y - (($data [$i - 1] / $max) * 200);                      //设置网站访问量的纵坐标值
        $y_ht = $y - (($data [$i] / $max) * 200);                          //获取每月网站访问量数的纵坐标值
        imageline ( $im, $x * ($i * 2 - 1) + 30, $y_lt, $x * (($i + 1) * 2 - 1) + 30, $y_ht, $red ); //绘制网站访问量折线图
}
for($i = 1; $i < 13; $i ++) {
        $r1 = round ( (($data [$i - 1]) / $max) * 100, 2 );
        imagestring ( $im, 2, $x * ($i - 1) * 2 + 40, $y + 11, $month [$i - 1], $black );   //输出月份的值
        imagestring ( $im, 2, $x * ($i - 1) * 2 + 36, $y + 25, $r1 . "%", $red );           //输出网站访问量的百分比
}
imagepng ( $im );
imagedestroy ( $im );                                                      //释放图像资源
```

秘笈心法

心法领悟 311：折线图中数据的获取方式。

在本实例中，将表单提交的数据直接作为折线图分析的数据。另外，还可以以从数据库中读取的数据作为折线图分析的数据，其实现方法与将数据库中的数据输出到页面中是相同的。

实例 312　GD2 函数柱形图分析编程词典满意度调查

光盘位置：光盘\MR\06\312

初级

趣味指数：★★★★☆

实例说明

在网站中，经常需要对相关的信息进行调查统计，然后根据访问者的投票结果制订相关计划，为了能够更直观地查看访问者的投票结果，本实例采用柱形图显示编程满意度的投票结果，其运行结果如图 6.45 所示。

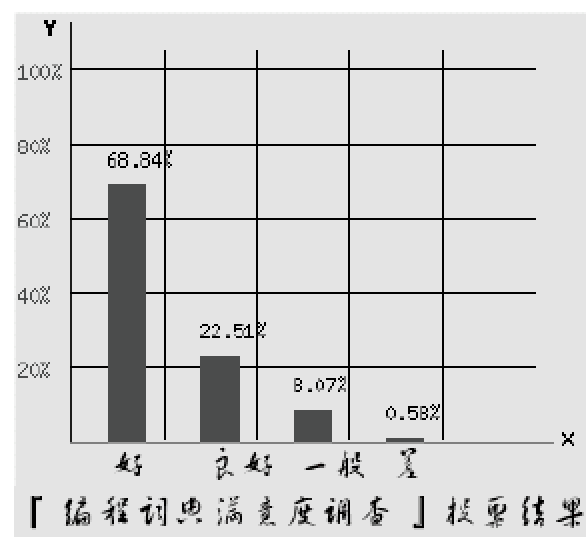


图 6.45　GD2 函数柱形图分析编程词典满意度调查

关键技术

通过 GD2 函数创建柱形图，关键是应用 imagefilledrectangle()函数完成柱形图的绘制及填充。

imagefilledrectangle()函数，在图像中绘制一个用指定颜色填充的矩形，其语法如下：

```
bool imagefilledrectangle ( resource image, int x1, int y1, int x2, int y2, int color )
```

参数说明：

image：图像资源。

x1：柱形图左上角横坐标；

y1：柱形图左上角纵坐标；

x2：柱形图右下角横坐标；

y2：柱形图右下角纵坐标；

color：填充颜色。

在本实例中，将从数据库中读取的数据以柱形图的形式进行展示，实现柱形图与数据库中数据的完美结合。

设计过程

（1）创建 index.php 页面，编写<img>标签输出 img.png 图像，即输出柱形图统计结果。

（2）创建 img.php 文件，从数据库中读取数据，将从数据库中读取的数据以柱形图形式进行输出，其关键代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
include ("conn/conn.php");
$query = mysql_query ( "select sum(ticket) as total from tb_comment" );
$info = mysql_fetch_array ( $query );
$max = $info [total];                                                    //所有投票的累加和
$sql = mysql_query ( "select * from tb_comment" );
$result = mysql_fetch_array ( $sql );                                    //检索数据信息
$im = imagecreate ( 350, 300 );                                          //创建画布
$green = imagecolorallocate ( $im, 233, 240, 151 );                      //设置颜色值
$black = imagecolorallocate ( $im, 0, 0, 0 );
$red = imagecolorallocate ( $im, 255, 0, 0 );
$blue = imagecolorallocate ( $im, 0, 0, 255 );
imageline ( $im, 30, 230, 290, 230, $blue );                             //设置 X 轴横坐标
imageline ( $im, 30, 5, 30, 230, $blue );                                //设置 Y 轴纵坐标
imagestring ( $im, 3, 295, 222, "X", $black );                           //输出字符 X
imagestring ( $im, 3, 16, 1, "Y", $black );                              //输出字符 Y
$l = 190;
$k1 = 30;
$k2 = 280;
for($j = 0; $j < 5; $j ++) {
 imageline ( $im, $k1, $l, $k2, $l, $black );                            //设置 X 轴网格线横坐标
```

```
 $l = $l - 40;
}
$f = 80;
$z1 = 20;
$z2 = 228;
for($j = 0; $j < 4; $j ++) {
 imageline ( $im, $f, $z1, $f, $z2, $black );                          //设置 Y 轴网格线纵坐标
 $f = $f + 50;
}
$l = 185;
for($j = 1; $j < 6; $j ++) {
 imagestring ( $im, 2, 2, $l, 20 * $j . "%", $red );
 $l = $l - 40;
}
$x = 45;
$y = 230;
$x_width = 20;
$y_ht = 0;
do {
        if ($result [comment] == "好") {
                $comment = "好";                                       //定义输出字体串
        }
        if ($result [comment] == "良好") {
                $comment = "良好";
        }
        if ($result [comment] == "一般") {
                $comment = "一般";
        }
        if ($result [comment] == "差") {
                $comment = "差";                                       //定义输出字体串
        }
        $fnt = "Font/FZHCJW.TTF";                                      //定义字体
        $y_ht = round ( ($result [ticket] / $max) * 200, 2 );          //设置投票结果所占百分比
        $y_ht1 = round ( ($result [ticket] / $max) * 100, 2 );         //设置投票结果所占百分比
        imagefilledrectangle ( $im, $x, $y, $x + $x_width, $y - $y_ht, $red );   //绘制并填充柱形图
        imagestring ( $im, 2, $x + $x_width - 20, $y - $y_ht - 20, $y_ht1 . "%", $black );   //设置投票结果所占百分比
        imageTTFText ( $im, 15, 0, $x + 2, $y + 20, $black, $fnt, $comment );   //写 TTF 文字到图中
        $x += ($x_width + 30);                                         //设置两柱形图之间的宽度为 30 像素
} while ( $result = mysql_fetch_array ( $sql ) );
$text = "『 编程词典满意度调查 』投票结果";
imageTTFText ( $im, 15, 0, 10, $y + 50, $black, $fnt, $text );         //写 TTF 文字到图中
imagedestroy ( $im );                                                  //释放图像资源
?>
```

秘笈心法

心法领悟 312：注意编码格式的选择。

在本实例中运用的是 utf-8 编码格式，所以在图像中输出中文字符串时可以直接应用 imageTTFText()函数；但是，如果页面使用的是 gb2312 编码格式，那么要输出中文字符串时，就必须将中文字符串转换成 utf-8 编码格式，否则将输出乱码。

实例 313 GD2 函数饼形图分析图书市场的份额

光盘位置：光盘\MR\06\313

高级

趣味指数：★★★★☆

实例说明

在调查某类商品的市场占有率时，最好的显示方式就是使用饼形图，通过饼形图可以直观地看到某类产品的不同品牌在市场中的占有比例。运行本实例，通过饼形图将不同语言的软件图书在市场中的占有率显示出来，运行结果如图 6.46 所示。

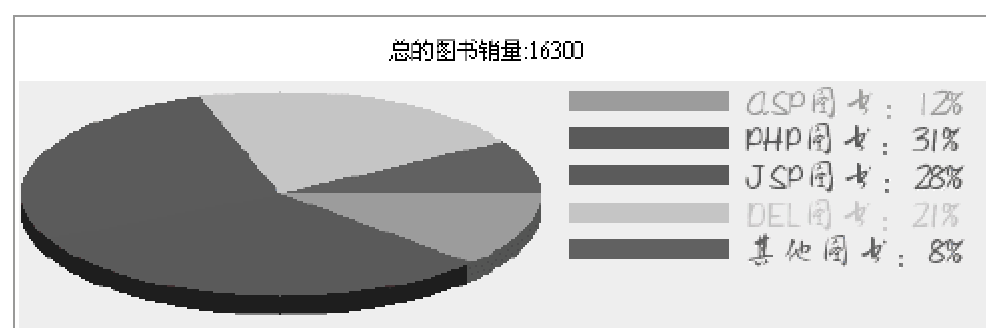


图 6.46　饼形图分析软件图书市场占有率

关键技术

（1）绘制椭圆弧线使用 GD2 函数库中的 imagefilledarc()函数，该函数在指定图像标识所代表的图像中以图像左上角为坐标点绘制一椭圆弧。如果成功返回 TRUE，否则返回 FALSE，函数的语法如下：

```
bool imagefilledarc ( resource image, int cx, int cy, int w, int h, int s, int e, int color, int style )
```

参数说明：

image：图像标识。

cx：图像左上角的横坐标。

cy：图像左上角的纵坐标。

w：椭圆的宽度。

h：椭圆的高度。

s：以角度指定椭圆的起始点。

e：以角度指定椭圆的结束点。

color：颜色标识。

style：可以是 IMG_ARC_PIE、IMG_ARC_CHORD、IMG_ARC_NOFILL、IMG_ARC_EDGED 按位或（OR）后的值。IMG_ARC_PIE 和 IMG_ARC_CHORD 是互斥的，IMG_ARC_CHORD 只是用直线连接了起始和结束点，IMG_ARC_PIE 则产生圆形边界（如果两个都用，IMG_ARC_CHORD 生效）；IMG_ARC_NOFILL 指明弧或弦只有轮廓，不填充；IMG_ARC_EDGED 指明用直线将起始和结束点与中心点相连，和 IMG_ARC_NOFILL 一起使用是绘制饼状图轮廓的好方法（而不用填充）。

（2）应用绘制柱形图的方法，通过柱形图展示不同颜色代表的数据。有关柱形图的应用请参考实例 312。

（3）充分展示数据库中数据与图形结合的方法。分别运用 for 语句和 while 语句对数据库中的数据进行循环输出。

设计过程

（1）创建 index.php 文件，连接数据库，执行查询语句，获取数据库中总的图书销量值，并通过<img>标签调用图像处理文件 caky_img.php。

（2）创建 caky_img.php 文件，通过饼形图和柱形图联合分析各语言图书的市场份额。首先，获取表单中传递的总的图书销量值，并计算各部门图书销量值占总图书销量的比例。然后，根据比例值创建饼形图。最后，通过柱形图输出不同的颜色区块代表的图书，其关键代码如下：

```
<?php
include ("conn/conn.php");                                              //连接数据库
$query = mysql_query ( "select product_count from tb_data " );         //执行查询语句
$data = array ();                                                       //定义空数组
while ( $myrow = mysql_fetch_array ( $query ) ) {
 $data [] = $myrow [product_count];                                    //将查询结果写入到数组中
}
$widths = 580;                                                          //定义图像宽度
$heights = 150;                                                         //定义图像高度
$width = 280;                                                           //定义饼形图宽度
$height = 130;                                                          //定义饼形图高度
$angle = array ();                                                      //角度
$total = 0;
```

```
for($i = 0; $i < count ( $data ); $i ++) {
        if (! is_numeric ( $data [$i] ))                                  //判断数组中的值是否为空
            die ( "1" );
        $total += $data [$i];
}
for($i = 0; $i < count ( $data ); $i ++) {
        array_push ( $angle, round ( 360 * $data [$i] / $total ) );      //定义饼形图的角度值
}
$image = imagecreate ( $widths, $heights );                                //创建饼形图的画布
$white = imagecolorallocate ( $image, 0xEE, 0xEE, 0xEE );                  //定义背景颜色
$color = array (array (imagecolorallocate ( $image, 0x97, 0xbd, 0x00) );   //定义图像的颜色值
$rounds_x = $width / 2;                                                    //原线
for($h = $height / 2 + 5; $h > $height / 2 - 5; $h --) {
        $start = 0;
        $stop = 0;
        for($i = 0; $i < count ( $data ); $i ++) {
            $start = $start + 0;
            $stop = $start + $angle [$i];
            $color_i = fmod ( $i, 10 );
            imagefilledarc ( $image, $rounds_x, $h, $width, $height - 20, $start, $stop, $color [1] [$color_i], IMG_ARC_PIE );
                                                                                //创建饼形图
            $start += $angle [$i];
            $stop += $angle [$i];
        }
}
for($i = 0; $i < count ( $data ); $i ++) {
        $start = $start + 0;
        $stop = $start + $angle [$i];
        $color_i = fmod ( $i, 10 );
        imagefilledarc ( $image, $rounds_x, $h, $width, $height - 20, $start, $stop, $color [0] [$color_i], IMG_ARC_PIE );
        $start += $angle [$i];
        $stop += $angle [$i];
}
$fnt = "Font/FZHCJW.TTF";                                                  //定义字体
$m = - 1;
$result = mysql_query ( "select * from tb_data " );                        //读取数据库中存储的数据
while ( $myrow = mysql_fetch_array ( $result ) ) {                         //while 循环输出数据库中数据
        $m ++;
        imagefilledrectangle ( $image, 295, 15 + 20 * $m, 380, 5 + 20 * $m, $color [0] [$m] );   //绘制并填充柱形图
        $comment = $myrow [product_name] . "：  " . number_format ( $myrow [product_count] / $_GET [counts], 2 ) * 100 . "%";
                                                                           //输出柱形图
        imageTTFText ( $image, 15, 0, 390, 18 + 20 * $m, $color [0] [$m], $fnt, $comment );   //写 TTF 文字到图中
}
imagepng ( $image );
imagedestroy ( $image );
?>
```

（3）编写 conn 文件夹，定义 conn.php 文件，连接 MySQL 数据库服务器，连接 db_database06 数据库，并设置数据库编码格式为 utf-8。

秘笈心法

心法领悟 313：for 循环语句和 while 循环语句在输出相同数据时的区别。

同时应用 for 循环语句和 while 循环语句输出数据库中存储的数据，如果定义变量$i 的初始值为 0，那么在 for 循环语句中，$i 的第一个值是 0；而在 while 循环语句中，$i 的第一个值是 1。这就是我们在本实例中为什么将$m 的初始值设置为-1 的原因。只有这样才能保证$m 与$i 的值是统一的，以获取相同的颜色值。如果将$m 与$i 的初始值都设置为 0，那么输出的饼形图区块的颜色将与柱形图区块的颜色不同。

第 7 章

面向对象

- 类的定义
- 类的访问修饰符
- 类的继承
- 抽象类和接口
- 类的多态
- 常用关键字
- 魔术方法
- 设计模式

7.1 类的定义

采用 PHP 面向对象的编程方式，首先需要对类进行定义。类是对象的模板，所以要定义规范、合理的类结构首先应该明确类应该具有的属性和方法，本节将通过具体的实例讲解类的定义方法。

实例 314 使用类的属性保存数据库连接参数

光盘位置：光盘\MR\07\314

初级

趣味指数：★★★

实例说明

运行本实例，如图 7.1 所示。首先在图中的文本框内输入连接 MySQL 数据库服务器所需的必要参数，然后单击“连接”按钮即可建立与指定的 MySQL 数据库服务器的连接，并将连接结果打印在页面中。

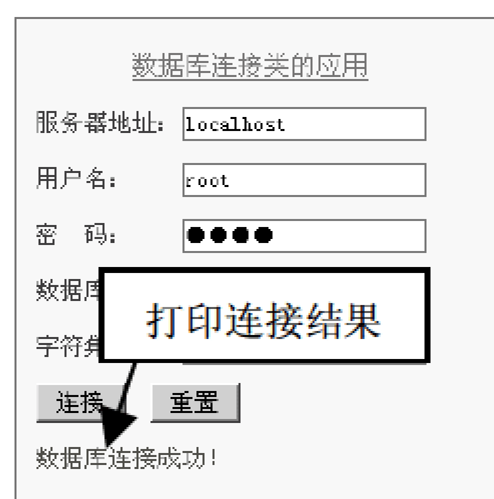


图 7.1　数据库连接成功

关键技术

类是对事物的抽象，在对类进行定义时，合理地设置类的属性和方法比较重要。例如在本实例所应用的数据库连接类中，将数据库服务器地址、数据库服务器用户名和密码、数据库名称以及数据库字符集作为类的属性保存，而将数据库连接的实现过程定义为类的方法，这是因为数据库连接参数在类的生命周期内基本没有变化，而且属于特性范围，所以定义为类的属性，而数据库连接方法属于类的动作或功能，所以定义成类的方法。

在 PHP 5.0 以前的版本中，类属性使用关键字 dim 声明，通过该关键字声明的类属性可以在类声明周期的任何范围内被调用。而在 PHP5.0 以后版本中引入了 public、protected 和 private 关键字来对类中的属性或方法进行声明，这样根据不同的关键字将类的属性或方法划分为公有成员、保护成员和私有成员 3 种，其中不同关键字修饰的类成员的说明和区别如下：

☑ public，表示该成员可以在任何范围被调用，包括类体内部、该类的子类和类实例的对象。

☑ protected，表示该方法可以在类中或该类的子类中被调用。

☑ private，表示该类只能在类体内被调用。

说明：如果没有对类中方法指定任何访问权限，PHP 会默认该方法的访问权限为 public。

设计过程

实现本实例，首先需要定义数据库连接类，然后建立数据库参数录入表单来指定数据库连接时所需要的连接参数，下面具体讲解该数据库连接类的设计和实现过程。

（1）定义数据库连接类，并通过类的属性保存连接 MySQL 数据库所需的参数，该类的具体实现代码如下：

```
class ConnDb
{
    private $host;                                              //MySQL 数据库服务器地址
    private $username;                                          //用户名
    private $password;                                          //密码
    private $charset;                                           //数据库编码
    private $dbname;                                            //数据库的名称
    public function __construct ($host, $username, $password, $dbname, $charset = 'utf-8')     //构造方式实现类的初始化
    {
        $this->host = $host;                                    //初始服务器地址
        $this->username = $username;                            //初始用户名
        $this->password = $password;                            //初始用户密码
        $this->dbname = $dbname;
        $this->charset = $charset;                              //初始数据库字符集
    }
    public function getConnId ()                                //数据库连接方法
    {
        @$connId = mysql_connect($this->host, $this->username, $this->password);        //获得数据库连接句柄
        @mysql_select_db($this->dbname, $connId);
        @mysql_query('set names ' . $this->charset);            //设置字符集
        return $connId;                                         //返回连接句柄
    }
}
```

上述代码中，定义了$host、$username、$password、$charset 和$dbname 5 个私有属性，分别用来保存数据库服务器的主机地址或名称、数据库服务器用户名、用户密码、字符集和要连接的数据库名，并通过构造方法__construct()实现数据库连接参数的初始化，最后定义 getConnId()方法返回数据库连接句柄。

（2）建立数据库连接参数录入表单，该过程通过 HTML 语言的表单标签实现，具体实现代码请详见本书附带光盘。

（3）当单击数据库参数录入表单中的“连接”按钮时，将会使用如下代码判断是否成功连接数据库：

```
if(isset($_POST['host'])){
    require 'ConnDb.php';                                       //包含数据库连接类文件
    $connDb = new ConnDb($_POST['host'], $_POST['username'], $_POST['password'], $_POST['dbname'], $_POST['charset']); //实例数据库连接类
    echo !$connDb->getConnId()?'数据库连接失败！':'数据库连接成功！';           //打印连接结果
}
```

上述代码，首先通过 isset($_POST['host'])判断用户是否已经提交了表单，如果是，则使用 require()语句包含数据库连接类，然后使用 new 关键字实例该类，并通过调用类中的 getConnId()方法来返回数据库连接状态，最后打印出连接结果。

秘笈心法

心法领悟 314：用@屏蔽运行期错误。

在程序运行时，有时会难以避免地出现运行期错误并在页面中打印错误信息，这样不仅会使页面的友好性大打折扣，而且还会为程序带来极大安全隐患。使用 try/catch 语句可以屏蔽运行期错误，不过需要书写的代码量相对较大。PHP 提供了@关键字来屏蔽运行期错误，只要在可能发生运行期错误的代码前加上“@”即可，例如本实例用来屏蔽连接数据库失败时的错误提示代码如下：

```
@$connId = mysql_connect($this->host, $this->username, $this->password);        //获得数据库连接句柄
```

实例 315	数据库连接类中定义数据库连接方法 光盘位置：光盘\MR\07\315	初级 趣味指数：★★★

实例说明

使用 PHP 的 MySQL 函数库管理 MySQL 数据库时，首先需要获得数据库连接句柄，然后才能进一步进行增、删、改、查操作。运行本实例，首先在图 7.2 所示的表单中输入数据库连接参数，然后单击“连接”按钮，

如果成功连接上 MySQL 数据库则会在页面中打印出数据库连接句柄。



图 7.2　打印连接句柄

关键技术

在 PHP 面向对象的编程方式中，若函数被定义在类中，就称之为成员方法，成员方法的功能应该匹配该类的特定功能。

类中定义的成员方法不能出现重名的情况，并且需要注意以下几点：

☑　函数名称不区分大小写，如 myfun()与 Myfun()指的是同一个函数。

☑　函数的参数没有限制，可以定义任意个需要的参数。

☑　函数名称理论上可以使用汉字，但汉字是双字节字符，为了避免出现无法预知的问题，尽量不使用。

设计过程

（1）定义数据库连接类，在类中定义数据库连接的方法 getConnId()，实现该过程的代码如下所示：

```
public function getConnId ()                                                    //数据库连接方法
{
    @$connId = mysql_connect($this->host, $this->username, $this->password);    //获得数据库连接句柄
    @mysql_select_db($this->dbname, $connId);
    @mysql_query('set names ' . $this->charset);                                //设置字符集
    return $connId;                                                             //返回连接句柄
}
```

在上述方法中，首先通过函数 mysql_connect()获得数据库连接句柄，然后使用函数 mysql_select_db()选择要连接的数据库，并使用函数 mysql_query()执行查询语句来设置数据库的字符集。

（2）实例数据库连接类，并打印连接句柄，实现代码如下所示：

```
$connDb = new ConnDb($_POST['host'], $_POST['username'], $_POST['password'], $_POST['dbname'], $_POST['charset']);    //实例数据库连接类
echo !$connDb->getConnId()?'数据库连接失败！':'数据库连接成功！<br/>连接句柄为'.$connDb->getConnId();                  //打印连接句柄
```

秘笈心法

心法领悟 315：通过 set names utf8 语句动态设置数据库的字符集为 utf-8 编码。

在对数据库内容进行检索输出时，有时可能会出现乱码现象，一般都是由数据库字符集设置存在偏差所致，为了避免上述情况的发生，可以在执行数据库查询语句前执行“set names 编码”语句，在开发项目时，为了便于国际化，建议应用 utf-8 编码。

注意：设置 MySQL 数据库为 utf-8 编码所执行的语句应该为“set names utf8”，而非“set names utf-8”，即“utf”和“8”之间没有“-”。

实例 316　数据统计类中定义求数值平均值的方法

初级

光盘位置：光盘\MR\07\316

趣味指数：★★★

实例说明

运行本实例，如图 7.3 所示。首先在图中文本框中输入一组数字并用半角逗号分割，然后单击“计算”按钮，将在页面中打印出该组数字的平均值。

请输入一组数字，要求用半角逗号分割

10, 20, 30　计算

该组数字的平均数为20

图 7.3　计算一组数的平均值

关键技术

实现本实例时，由于该类只包含一个用于计算平均数的方法，所以在实现该类时将求平均数的方法定义为静态方法，这样在不实例该类的情况下通过类名即可直接调用该方法。

在 PHP 面向对象编程方式中，静态方法隶属于某个类，但不受类的束缚，可以直接被外部访问。与定义类中普通方法相比，定义静态方法时需要使用 static 关键字进行修饰，其定义格式如下所示：

```
public static function  方法名()
{
        //该方法实现的功能
}
```

在类体内部调用该方法时，使用 self 关键字加“::”调用，在类外直接使用类名加“::”引用。

设计过程

（1）建立数据统计类 Stat，并在类体内定义静态方法 getAvg()用来实现求多个数字的平均值，其实现代码如下所示：

```
class Stat
{
    public static function getAvg ($arryNum)                //定义静态方法来求平均值
    {
        $totalNum = count($arryNum);                        //获得数字总数
        if ($totalNum == 0) {                               //如果数字个数为 0 则返回 null
            return null;
        } else {
            $sum = 0;                                       //用变量$sum 保存所有数字的和
            for ($i = 0; $i < $totalNum; $i ++) {           //通过循环计算所有数字的和
                $sum += $arryNum[$i];
            }
            return $sum / $totalNum;                        //返回平均数
        }
    }
}
```

上述代码用于实现求平均数的 getAvg()方法中，为了能够实现求任意个数数字的平均数，将数组作为参数，这样只要求得该数组所有元素的平均值即可，在具体实现过程中，首先使用函数 count()计算数组元素的个数，然后使用 for 循环语句计算数组中所有元素的和，最后使用 return 语句返回所求得的平均数。

（2）建立数字录入表单，当用户单击表单中的“计算”按钮时，将通过如下代码计算在表单中录入数字的平均数：

```
if(isset($_POST['nums']) && trim($_POST['nums'])!=''){         //判断是否已经提交表单
    require 'Stat.php';                                        //包含数据统计类
    $arrayNum = explode(',', $_POST['nums']);                  //将提交数字保存到数组中
    echo '该组数字的平均数为' . Stat::getAvg($arrayNum);        //调用统计类的静态方法 getAvg()求平均数
}
```

上述代码首先判断用户是否提交了表单，如果是，则首先使用 require 语句包含数据统计类，然后使用函数 explode()用半角逗号分割提交的数字字符串，最后调用数据统计类的静态方法 getAvg()计算所录入数字的平均数。

说明： 函数 explode()用于实现将字符串以指定的字符或子串分割，并将分割结果保存到数组中。

秘笈心法

心法领悟 316：为了提高代码运行效率，建议将 count()函数放在循环体外。

本实例计算数组中所有元素的和可以有两种写法，分别如下：

☑ 将函数 count()写在 for 循环体内：

```
for($i=0; $i<count($arrayNum); $i++){}
```

☑ 将函数 count()写在 for 循环体外：

```
$count = count($arrayNum);
for($i=0; $i<$count; $i++){}
```

查看上述两种写法可知，第一种写法在执行每次循环时都需要执行一次函数 count()，而第二种写法只在执行循环前执行一次函数 count()，所以在程序开发过程中，建议采用第二种写法，可以提高程序运行效率。

实例 317 使用重载实现不同类型数据的运算

光盘位置：光盘\MR\07\317

初级

趣味指数：★★★

实例说明

本实例通过使用类的方法重载机制，实现不同类型数据的运算。运行本实例，分别如图 7.4 和图 7.5 所示。其中，图 7.4 展示的是两个数值型数据求和的计算结果，图 7.5 展示的是两个字符串连接的结果。

图 7.4　两个数字求和的结果

图 7.5　两个字符串连接的结果

关键技术

本实例主要应用 PHP 面向对象编程中的方法重载实现。通过方法重载机制，可以在子类中定义与父类同样的方法，然后对该方法内部功能代码进行重新编写，这样就实现了方法重载。通过方法重载可以扩展或弥补原父类方法的缺陷或不足，进一步扩展类中方法的功能。

例如，已经定义了较为复杂的汽车类 Car，并在其内部定义了变速行驶的方法，代码如下：

```
class Car
{
    public function changeSpeed()                    //变速的方法
    {
                                                     //实现手动变速的功能
    }
                                                     //其他方法
}
```

分析上述代码可知，其中的变速方法只能实现"手动变速"的功能，那么如果汽车经过改装可以实现自动变速了，是不是该 Car 类就需要完全重新编写呢？其实并不需要这样，只要编写该类的子类，然后在子类中重写 changeSpeed()方法即可，实现该过程的示例代码如下：

```
class NewCar extends Car
{
    public function changeSpeed()
    {
```

```
            if(条件 1){
                //实现手动变速的功能
            }else{
                //实现自动变速的功能
            }
        }
        //其他方法
}
```

经过对上述 changeSpeed()方法的重写，就可以使该方法既具有手动变速又具有自动变速的功能。从而可以缩短程序开发周期，节约维护时间。

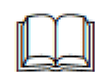
设计过程

（1）首先建立数据统计类父类，并在该类中定义统计方法 execStat()，父类所定义的统计方法只有计算两个数和的功能，其实现代码如下所示：

```
class StatParent                                                //定义数据统计父类
{
    public function execStat ($var0, $var1)                     //定义数据统计方法
    {
        return intval($var0 + $var1);                           //返回两个数的和
    }
}
```

（2）定义类 Stat，并使该类继承自 StatParent 类，在该类体内重写统计方法 execStat()，使该方法根据传递的参数类型种类来同时实现数值型的求和及字符串型的连接功能，代码如下：

```
require 'StatParent.php';
class Stat extends StatParent
{
    public function execStat ($var0, $var1)                     //重写数据统计方法
    {
        return is_numeric($var0) && is_numeric($var1) ? intval($var0 + $var1) : $var0 . $var1;   //如果为数字则计算数字的和,不是数字则连
                                                                  接两个参数
    }
}
```

 **说明：** 函数 is_numeric()用于判断指定的内容是否为数值型数据或数值型字符串。

（3）建立数据录入表单（该表单的实现代码请详见本书附带光盘），当用户在表单中录入完要统计的数据后，单击“统计”按钮，将通过如下代码调用数据统计类的 execStat()方法对用户提交的内容进行统计：

```
if(isset($_POST['txt1']) && $_POST['txt1']!=''){                //判断用户是否提交了表单
    require 'Stat.php';                                         //包含数值统计类
    $stat = new Stat();                                         //实例数值统计类
    echo '统计结果为：' . $stat->execStat($_POST['txt1'], $_POST['txt2']);   //输出统计结果
}
```

秘笈心法

心法领悟 317：使用条件运算符替代简单的 if 条件控制语句。

在编写程序时，可以使用条件运算符替代简单的 if 条件控制语句，PHP 中条件运算符的使用格式如下：

```
条件表达式?表达式 1:表达式 2
```

在执行条件运算符时，首先计算条件表达式的值，如果为真则执行表达式 1，反之执行表达式 2。

实例 318	使用$this 关键字调用汽车类自身的方法 光盘位置：光盘\MR\07\318	初级 趣味指数：★★★

实例说明

本实例通过$this 关键字在汽车类内部调用设置汽车颜色的方法和设置汽车品牌的方法来输出汽车的信息。

运行本实例，如图 7.6 所示，分别选中表单中的颜色和类型单选按钮，然后单击“提交”按钮即可在页面中打印出汽车的信息。

图 7.6　打印汽车信息

关键技术

在 PHP 面向对象编程方式中，可以使用$this 关键字对类自身的属性或方法进行调用。例如已经在 A 类中定义了方法 fun1()，在该类的 fun2()方法中可以使用如下方式对 fun1()进行调用：

```
class A
{
    private function func1()
    {
                    //fun1()实现的功能
    }
    public function fun2()
    {
        $this->fun1();      //调用 fun1()
    }
}
```

设计过程

（1）定义汽车类 Car。首先在该类中定义汽车颜色标识和汽车类别标识两个属性，然后使用构造函数对这两个属性进行初始化，并在类内定义 getColor()方法和 getType()方法，分别用来获取汽车颜色和汽车类型，最后使用 getInfo()方法打印出汽车信息。本实例的汽车类实现代码如下所示：

```
class Car                                                   //定义汽车类
{
    private $colorFlag;                                     //颜色标识
    private $typeFlag;                                      //类型标识
    public function __construct ($colorFlag, $typeFlag)     //构造方法
    {
        $this->colorFlag = $colorFlag;                      //颜色标识初始化
        $this->typeFlag = $typeFlag;                        //类型标识初始化
    }
    public function getColor ()                             //定义获取颜色方法
    {
        switch ($this->colorFlag) {                         //使用 switch 语句根据不用颜色获得标识和颜色
            case 0:
                $color = '红色';
                break;
            case 1:
                $color = '白色';
                break;
            case 2:
                $color = '黑色';
                break;
            default:
                $color = '宝石蓝';
        }
        return $color;
    }
    public function getType ()                              //定义获得汽车类型方法
    {
        switch ($this->typeFlag) {                          //根据类型标识获得汽车类型
```

```
            case 0:
                $type = '奔驰';
                break;
            case 1:
                $type = '宝马';
                break;
            case 2:
                $type = '奥迪';
                break;
            default:
                $type = '捷达';
        }
        return $type;
    }
    public function getInfo ()                                          //获得汽车信息
    {
        return '我的汽车是' . $this->getColor() . $this->getType();     //调用类内方法返回汽车信息
    }
}
```

（2）建立汽车属性选择表单，当用户单击表单的“提交”按钮后接收表单提交的颜色标识和类别标识，然后使用 new 关键字对汽车类进行实例化，最后使用实例化的汽车类的对象调用该类的 getInfo()方法打印汽车信息，其代码如下：

```
if(isset($_POST['color']) && isset($_POST['type'])){                    //判断是否提交了表单
    require 'Car.php';                                                  //包含汽车类
    $colorFlag = $_POST['color'];                                       //获得表单提交的颜色标识
    $typFlag = $_POST['type'];                                          //获得表单提交的类别标识
    $car = new Car($colorFlag, $typFlag);                               //对汽车类进行实例化
    echo $car->getInfo();                                               //打印汽车信息
}
```

秘笈心法

心法领悟 318：在类体内部使用$this 关键字可以调用私有、保护和公有成员。

PHP 面向对象编程方式中，在类体内部可以使用$this 关键字访问被 private、protected 和 public 所修饰的属性或方法。

实例 319　使用 self 关键字调用学生类自身的静态方法

光盘位置：光盘\MR\07\319

初级

趣味指数：★★★

实例说明

本实例首先定义学生类 Student，然后在类中分别定义学习和读书两个静态方法，同时在类中定义非静态的 getAllMethod()方法，通过该方法使用 self 关键字调用学习和读书这两个静态方法，在类外分别使用类名和实例的对象调用类中方法将输出如图 7.7 所示的结果。

直接使用类名调用类的静态方法：　学生可以学习、读书

通过类对象调用类中方法：　学生可以学习、读书

图 7.7　调用类中的静态方法

关键技术

PHP 面向对象的编程方式中，可以使用 self 关键字调用类中的静态属性或方法，调用格式分别如下所示：

```
class A
{
    private static $var;                                                //定义静态属性$var
```

```
    private static function fun(){                              //定义静态方法 fun()

    }
    public function main(){                                     //定义公有方法 main()
        self::$var;                                             //引用静态属性
        self::fun();                                            //引用静态方法
    }
}
```

从上述代码可知，在类体内部使用 self 关键字加“::”引用类中的静态属性或静态方法。

注意：使用 self 关键字调用类的静态属性时不要遗漏$，例如本实例中的引用方法为 self::$var，而非 self::var。

设计过程

（1）定义学生类，在类中定义学生学习的静态方法和学生读书的静态方法，同时定义非静态方法 getAllMethod()，并在该方法内部使用 self 关键字调用类中的静态方法返回学生可以完成的活动，代码如下所示：

```
class Student
{
    public static function study ()                             //定义学习方法
    {
        return '学习';
    }
    public static function read ()                              //定义读书方法
    {
        return '读书';
    }
    public function getAllMethod ()                             //定义 getAllMethod()来获得类中所有方法
    {
        return '学生可以' . self::study() . '、' . self::read();
    }
}
```

（2）使用 require 语句包含 Student.php 文件，然后分别使用类名和实例的对象调用类中的静态方法和 getAllMethod()方法打印学生可以进行的活动，代码如下所示：

```
<?php
require 'Student.php';                                          //包含 Student.php 文件
?>
<font color="red">
//直接使用类名调用类的静态方法
学生可以<?php echo Student::study()?>、<?php echo Student::read()?></font><br/>
<font color="blue">
//通过类对象调用类中方法
<?php
$student = new Student();                                       //对学生类进行实例化
echo $student->getAllMethod();                                  //调用学生类的 getAllMethod()方法
?>
```

秘笈心法

心法领悟 319：使用类名调用类中静态方法，将不执行构造方法。

通过类名可以直接调用类中的静态方法，那么此时是否会执行构造方法__construct()呢？请看如下代码：

```
class A
{
    public function __construct ()                              //构造方法
    {
        echo '构造方法';
    }
    public static function fun ()                               //静态的 fun()方法
    {
        echo 'fun()';
    }
}
A::fun();                                                       //通过类名调用静态的 fun()方法
```

运行上述代码将输出 fun()，而没有输出构造方法中的内容，而当使用 new 关键字对类进行实例化后将执行构造方法。

实例 320　调用汽车类中的刹车方法和颜色属性　初级

光盘位置：光盘\MR\07\320　趣味指数：★★★

实例说明

本实例主要实现通过实例后类的对象调用类的属性和方法。运行本实例，如图 7.8 所示，通过调用汽车类的颜色属性打印汽车颜色，通过调用汽车类的刹车方法执行刹车操作。

汽车的颜色为 — 红色
行驶过程中 — 汽车执行了刹车方法

图 7.8　通过对象调用类中属性和方法

关键技术

类实例后的对象不仅可以调用类的公有方法，而且可以调用类中的公有属性，其调用格式如下：

```
$object->属性;
$object->方法;
```

设计过程

（1）定义汽车类，在该类中定义汽车颜色属性和汽车刹车的方法，并通过构造方法对汽车颜色属性进行初始化，其实现代码如下所示：

```
class Car
{
    public $color;                              //汽车颜色属性

    public function __construct($color){        //构造方法
        $this->color = $color;                  //汽车颜色初始化
    }

    public function stop(){                     //定义汽车刹车方法
        return "汽车执行了刹车方法";
    }
}
```

注意：为了能够在类体外引用类的颜色属性，需要将颜色属性声明为公有属性。

（2）使用 require 语句在 index.php 文件中包含 Car.php 文件，并对汽车类进行实例化，同时指定汽车类的颜色属性值为红色，最后使用实例化的对象通过符号“->”调用汽车类的颜色属性和刹车方法，代码如下所示：

```
<?php
require 'Car.php';
?>
<font color="blue">
<?php
$color = '红色';
$car = new Car($color);
echo '汽车的颜色为 — '. $car->color.'<br/>';
echo '行驶过程中 — '. $car->stop();
?>
```

秘笈心法

心法领悟 320：使用$this 引用类的属性时，请注意“$”的位置。

引用类内的属性时，“$”应该放在 this 前，即$this->var，而非 this->$var。

实例 321 学生类中使用构造方法为学生信息初始化

光盘位置：光盘\MR\07\321

中级

趣味指数：★★★

实例说明

本实例采用二维数组模拟数据库存储学生信息，然后通过 for 循环语句实例学生类，并通过构造方法为学生类的属性赋初值，最后通过类的 getXxx()方法获得学生属性，并将学生信息在页面输出，如图 7.9 所示。

学号	姓名	年龄	住址
0312310	小明	16	北京西城区
0312311	小张	16	北京宣武区
0312312	小赵	17	北京海淀区

图 7.9　学生信息列表

关键技术

构造方法是类中的一个特殊函数，当创建一个类的实例时，将会自动调用类的构造方法。在 PHP 5.0 以前的版本中，构造方法名称与类名相同，而在 PHP 5.0 以后的版本中统一使用__construct()作为构造方法名称，使用构造方法应该注意以下两点：

☑ 构造方法没有返回值类型和返回值，这是因为构造方法是在创建对象时自动调用的，并不是一个独立的函数，因此不需要返回值。

☑ 构造方法的主要功能是实现对类的初始化工作。

设计过程

（1）首先建立二维数组$students 用来模拟数据库，代码如下所示：

```
$students = array(                                              //二维数组，模拟数据库
    0 => array('0312310', '小明', '16', '北京西城区') ,
    1 => array('0312311', '小张', '16', '北京宣武区') ,
    2 => array('0312312', '小赵', '17', '北京海淀区')
);
```

（2）定义学生类 Student，在该类中使用构造方法__construct()实现对学生信息的初始化，然后针对类的各个属性建立相应的 getXxx()方法来获取属性的值，代码如下：

```
class Student
{
    private $id;                                                //学生 ID
    private $name;                                              //学生名称
    private $age;                                               //学生年龄
    private $address;                                           //学生住址
    public function __construct ($id, $name, $age, $address)    //构造方法，对学生信息初始化
    {
        $this->id = $id;
        $this->name = $name;
        $this->age = $age;
        $this->address = $address;
    }
    public function getId ()                                    //获得学生 ID
    {
        return $this->id;
    }
    public function getName ()                                  //获得学生名称
    {
        return $this->name;
    }
```

```
    public function getAge ()                          //获得学生年龄
    {
        return $this->age;
    }
    public function getAddress ()                      //获得学生住址
    {
        return $this->address;
    }
}
```

（3）通过 for 循环语句遍历保存学生信息的$students 数组，遍历过程中将学生信息保存到 Student 学生类中，并通过该类实例的对象调用类中相应的 getXxx()方法，获取学生信息。实现该过程的代码如下：

```
<?php
require 'arrayDb.php';
require 'Student.php';
$count = count($students);
?>
<table border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#009933">
  <tr>
    <td width="100" height="22" bgcolor="#009933"><font color="#FFFFFF">学号</font></td>
    <td width="100" bgcolor="#009933"><font color="#FFFFFF">姓名</font></td>
    <td width="100" bgcolor="#009933"><font color="#FFFFFF">年龄</font></td>
    <td width="200" bgcolor="#009933"><font color="#FFFFFF">住址</font></td>
  </tr>
<?php
for ($i=0; $i<$count; $i++){
    $stu = $students[$i];
    $student = new Student($stu[0], $stu[1], $stu[2], $stu[3]);
?>
  <tr>
    <td height="22" bgcolor="#FFFFFF"><?= $student->getId()?></td>
    <td bgcolor="#FFFFFF"><?= $student->getName()?></td>
    <td bgcolor="#FFFFFF"><?= $student->getAge()?></td>
    <td bgcolor="#FFFFFF"><?= $student->getAddress()?></td>
  </tr>
<?php
}
?>
```

秘笈心法

心法领悟 321：PHP 5.0 中构造方法名称以双下划线开头。

在使用 PHP 5.0 的构造方法时，一定要注意构造方法__construct()是以双下划线开头，而非单下划线。

实例 322　圆类中使用 const 关键字定义圆周率类常量

中级

光盘位置：光盘\MR\07\322

趣味指数：★★★

实例说明

本实例实现的功能是根据圆的半径计算圆的面积，运行本实例，如图 7.10 所示，首先在图中的文本框内输入要计算的圆面积的半径，然后单击“计算”按钮即可计算出圆的面积，并将结果打印在页面中。

计算圆面积：
请输入圆的半径：2　计算
该圆的面积为：12.56

图 7.10　计算圆面积

关键技术

本实例的关键是通过类常量保存圆的半径，然后在类的方法中对该常量进行调用。下面介绍 PHP 面向对象编程方式中类常量的定义和引用方法。

大家知道 PHP 的全局常量可以使用函数 define()定义，例如：

```
define('PI', '3.17');                                //定义常量 PI
$radius = 2;                                         //指定圆半径
echo '圆面积为：' . PI * pow($radius, 2);            //计算圆面积
```

在上述代码中首先使用函数 define()定义圆周率常量 PI，然后指定圆半径为 2，最后在计算圆面积的表达式中调用常量 PI 计算圆面积。

而在 PHP 5.0 以后的面向对象编程方式中，引入了类常量的概念，类常量属于类本身，而且区分大小写，在类体内与静态成员类似，都使用 self 关键字加“::”对类常量进行引用，例如本实例所定义的圆类中通过如下代码使用类常量计算圆面积：

```
class Circle
{
    const PI = '3.14';                               //定义圆周率为 3.14，也可以使用 pi()函数
    ...//省略部分代码
    public function getArea ()
    {                                                //获取圆面积
        return self::PI * pow($this->radius, 2);
    }
}
```

从上述代码可知，类常量使用 const 关键字声明，与全局常量类似，习惯用大写字母表示，在类的方法中使用 self::调用。

设计过程

（1）定义圆类，首先将圆周率保存在类常量 PI 中，然后定义私有方法$radius 用于保存圆的半径，最后定义类的公有方法 getArea()来计算圆的面积，实现代码如下所示：

```
class Circle
{
    const PI = '3.14';                               //定义圆周率为 3.14，也可以使用 pi()函数
    private $radius;                                 //圆半径
    public function __construct ($radius)
    {                                                //构造方法，实现对圆半径初始化
        $this->radius = $radius;
    }
    public function getArea ()
    {                                                //获取圆面积
        return self::PI * pow($this->radius, 2);
    }
}
```

（2）建立圆半径录入表单（该过程的实现代码请详见本书附带光盘），在表单中录入圆半径后，单击“计算”按钮即可计算圆面积并打印计算结果，代码如下。

```
if (isset($_POST['radius']) && trim($_POST['radius'])!=''){   //判断是否提交了表单
    require_once 'Circle.php';                                 //包含圆类
    $circle = new Circle($_POST['radius']);                    //对圆类进行实例化
    echo '该圆的面积为：'.$circle->getArea();                  //打印计算结果
}
```

秘笈心法

心法领悟 322：使用函数 pow()计算指定数值的 n 次方幂。

在 PHP 编程中，经常会计算某个数的 n 次方幂，PHP 提供了函数 pow()用于计算某个数的 n 次方幂，该函数的语法格式如下：

```
number pow (number base, number exp)
```

参数说明：

base：要计算的数值；

exp：幂数。

该函数用于计算数值 base 的 exp 次幂的值，如果不能计算，则会发出一条警告并返回 false 值。

7.2　类的访问修饰符

在 PHP 5.0 以前的版本中，所有类成员被视为公有成员，即在类体内、类的子类中以及类实例后的对象均可被调用，而在 PHP 5.0 以后的版本中，引入了私有成员、保护成员和公有成员的概念，从而使得 PHP 面向对象的编程方式更为成熟化。

实例 323	汽车类使用 public 关键字定义汽车的行驶方法 光盘位置：光盘\MR\07\323	中级 趣味指数：★★★

实例说明

本实例用于展示在类体外、类内部的方法中和类的子类中调用被 public 关键字所修饰的方法的结果，如图 7.11 所示。

（1）通过汽车类对象调用汽车类的行驶方法的结果：
行驶
（2）通过汽车类的getStatus()方法调用汽车行驶方法的结果：
汽车目前正在行驶
（3）通过汽车类的子类调用汽车类的汽车行驶方法的结果：
小汽车行驶

图 7.11　被 public 关键字修饰的方法在不同范围内被调用

关键技术

在 PHP 5.0 面向对象的编程中，被关键字 public 修饰的方法称为公有方法，公有方法可以被类实例的对象在类内部以及类的子类中调用。

设计过程

（1）首先建立汽车类 Car，并在该类内部定义汽车行驶方法 run()以及获得汽车状态的方法 getStatus()，然后定义类 SmallCar，使该类继承自类 Car，并在该类中定义 smallCarRun()方法。实现该过程的代码如下所示：

```
class Car                                          //定义汽车类 Car
{
    public function run ()                         //定义行驶方法
    {
        return '行驶';
    }
    public function getStatus ()                   //定义返回汽车状态方法
    {
        return '汽车目前正在' . $this->run();
    }
}
//定义小汽车类 SmallCar，并使其继承汽车类 Car
class SmallCar extends Car
{
    public function smallCarRun ()                 //定义小汽车行驶方法
    {
        return '小汽车' . $this->run();
    }
}
```

（2）打印在不同范围内调用汽车类的行驶方法的结果，代码如下所示：

```
require 'Car.php';
$car = new Car();
echo '（1）通过汽车类对象调用汽车类的行驶方法的结果：<br/>';
echo $car->run() . '<br/>';
echo '（2）通过汽车类的 getStatus()方法调用汽车行驶方法的结果：<br/>';
echo $car->getStatus() . '<br/>';
echo '（3）通过汽车类的子类调用汽车类的汽车行驶方法的结果：<br/>';
$smallCar = new SmallCar();
echo $smallCar->smallCarRun() . '<br/>';
```

秘笈心法

心法领悟 323：合理使用 public 关键字修饰类体内的方法。

在定义类的时候，虽然使用 public 关键字可以使方法在类的生命周期的任何范围内被访问，但是不要将类中的所有方法都定义为公有，这样可以有效提高程序的安全性和商业化。

实例 324 使用 private 关键字定义汽车的颜色属性

光盘位置：光盘\MR\07\324

中级

趣味指数：★★★

实例说明

本实例主要通过在不同范围内调用类中私有属性来演示私有成员的作用范围。运行本实例，如果通过类实例后的对象直接调用类的私有属性，将在页面打印如图 7.12 所示的错误信息，如果通过类中方法调用类的私有属性将输出汽车的颜色。

通过类实例的对象调用类的私有属性的结果：

Fatal error: Cannot access private property Car::$color in **D:\AppServ\www\MR\07\011\index.php** on line **17**

图 7.12　通过对象调用类中私有属性

在类中的方法中调用类的私有属性的结果：

汽车颜色是：红色

图 7.13　通过类中方法调用类的私有属性

关键技术

在 PHP 5.0 面向对象的编程中，被关键字 private 修饰的成员称为私有成员，私有成员可以在类体内被调用，但不可以被类实例的对象和类的子类所调用。

设计过程

（1）首先定义汽车类 Car，在汽车类中定义私有属性$color，并通过构造方法为$color 属性进行初始化，最后定义 getColor()方法返回汽车颜色的值，该过程的实现代码如下：

```
class Car                                            //定义汽车类 Car
{
    private $color;                                  //汽车颜色
    public function __construct ($color)             //构造方法对类中属性初始化
    {
        $this->color = $color;                       //汽车颜色初始化
    }
    public function getColor ()                      //定义获得汽车颜色的方法
    {
        return $this->color;                         //返回汽车颜色
    }
}
```

（2）对汽车类进行实例化，并指定汽车颜色为红色，然后通过实例的对象调用汽车类中的 getColor()方法获得汽车颜色，该过程的实现代码如下所示。在下述代码中，被注释的代码用于实现通过类实例的对象调用类

中私有成员的结果，如果去掉注释将出现如图 7.12 所示的错误提示，从而说明类中的私有方法不能被类实例的对象所调用。

```
require 'Car.php';                                      //包含汽车类
$color = '红色';                                        //指定汽车颜色
$car = new Car($color);                                 //对汽车类实例化
//echo '通过类实例的对象调用类的私有属性的结果：<br/>';
//echo @'汽车颜色是：'.$car->color.'<br/>' or die('不能通过类的对象调用类中的私有方法<br/>');
echo '在类中的方法中调用类的私有属性的结果：<br/>';
echo '汽车颜色是：'.$car->getColor().'<br/>';           //打印汽车颜色
```

秘笈心法

心法领悟 324：合理定义类中的私有方法。

由于私有成员只可在类体内被访问，所以在定义类成员时，不准备在类体外及该类的子类中引用的方法，应该定义为私有方法。

如果打算在类体外获得某属性的值，建议将该属性定义为私有属性，然后再定义一个公有的 getXxx()方法，在该方法中使用 return 语句返回该属性的值，在类体外只要通过类实例的对象调用该方法即可获得相应私有属性的值。

实例 325	使用 protected 关键字定义汽车的保修年限 光盘位置：光盘\MR\07\325	中级 趣味指数：★★★

实例说明

本实例主要用于说明类中的保护成员在不同范围内被调用的结果。运行本实例，当使用类实例的对象调用类中的保护属性时，将出现如图 7.14 所示的错误提示，然后注释掉通过类对象调用类体内保护属性的代码，并分别通过在类体内和类的子类中调用类的保护属性，再次运行该实例，将出现如图 7.15 所示的页面。

通过类实例的对象直接调用类中的保护成员的结果

Fatal error: Cannot access protected property Car::$repairTime in **D:\AppServ\www\MR\07\012\index.php** on line **17**

图 7.14　通过对象调用类中保护成员

在类体内调用保护成员的结果
汽车保修年限为：3年
在类子类中调用保护成员的结果
小汽车保修年限为：5年

图 7.15　通过子类调用父类的保护成员

关键技术

在 PHP 5.0 面向对象的编程中，被关键字 protected 修饰的成员称为保护成员，保护成员可以在类体内以及类的子类中被调用，但不可以在类体外被调用。

设计过程

（1）定义汽车类 Car 以及该类的子类 SmallCar。在汽车类中定义汽车的保修时间属性，然后使用构造方法对该属性进行初始化，同时定义获得汽车保修时间的 getRepairTime()方法。在汽车类的子类 SmallCar 中，使用构造方法调用其父类的构造方法，从而实现对父类的保修时间属性进行初始化，最后在 SmallCar 类中定义 getSmallCarRepairTime()方法返回汽车的保修时间，实现该过程的代码如下：

```
class Car                                               //定义汽车类 Car
{
    protected $repairTime;                              //汽车保修时间属性
    public function __construct ($repairTime)           //构造方法对汽车的保修时间进行初始化
    {
```

```
        $this->repairTime = $repairTime;
    }
    public function getRepairTime ()                    //获得汽车保修时间的方法
    {
        return $this->repairTime;
    }
}
class SmallCar extends Car                              //定义汽车类的子类 SmallCar
{
    public function __construct ($repairTime)
    {
        parent::__construct($repairTime);               //调用父类构造方法对保修时间属性进行初始化
    }
    public function getSmallCarRepairTime ()            //调用父类的保修时间属性
    {
        return $this->repairTime;
    }
}
```

（2）使用 new 关键字对汽车类 Car 进行实例化，然后分别打印出在不同范围内调用汽车类保护属性 $repairTime 的结果，代码如下所示：

```
require 'Car.php';                                      //包含汽车类
$repairTime = 3;                                        //定义保修年限
$car = new Car($repairTime);                            //对汽车类实例化
//echo '通过类实例的对象直接调用类中的保护成员的结果<br/>';
//echo '汽车保修年限为：'.$car->repairTime.'年<br/>';
echo '在类体内调用保护成员的结果<br/>';
echo '汽车保修年限为：'.$car->getRepairTime().'年<br/>';
echo '在类子类中调用保护成员的结果<br/>';
$repairTime = 5;                                        //定义保修年限
$smallCar = new SmallCar($repairTime);
echo '小汽车保修年限为：'.$smallCar->getSmallCarRepairTime().'年<br/>';
```

秘笈心法

心法领悟 325：合理定义类中的保护成员。

如果类中的属性或方法，只能在类体中或类的子类中被访问，那么建议将该类定义为保护成员。

7.3 类的继承

继承性是类的主要特点之一，通过类的继承可以扩展父类的功能，本节将通过具体实例讲解 PHP 5.0 面向对象编程方式中继承的应用方法。

实例 326	苹果子类继承水果父类 光盘位置：光盘\MR\07\326	中级 趣味指数：★★★

实例说明

本实例首先定义父类水果类，并在该类中定义获得水果颜色和水果形状的方法，然后定义水果类的子类苹果类，在苹果类中只定义构造方法用于实现对类进行初始化，然后通过调用父类的方法获得苹果的颜色和形状。

运行本实例，如图 7.16 所示，首先在表单中输入苹果的颜色和形状属性，然后单击“提交”按钮即可在页面中打印出苹果的颜色和形状属性。

请输入苹果的颜色：红色
请输入苹果的形状：圆形
提交
我看见的苹果是：红色，圆形的

图 7.16　打印苹果颜色和形状属性

关键技术

在 PHP 面向对象编程方式中，可以使用 extends 关键字继承指定的父类，如下所示：

```
class Parent1                                            //定义父类 Parent1
{
        //父类的成员
}
class Child extends Parent1                              //定义子类 Child
{
        //子类的成员
}
```

通过上述继承关系，即可在子类中使用 parent 关键字加“::”操作符调用父类的公有方法或属性，或者重写非 final 关键字修饰的方法。

设计过程

（1）定义水果类 Fruit，在该类中定义颜色属性$color 和形状属性$shape，并通过构造方法对这两个属性进行初始化，然后定义用于获得这两个属性的 getColor()方法和 getShape()方法，代码如下所示：

```
class Fruit                                              //定义水果类
{
    private $color;                                      //颜色属性
    private $shape;                                      //形状属性
    public function __construct ($color, $shape)         //构造方法对水果类初始化
    {
        $this->color = $color;
        $this->shape = $shape;
    }
    public function getColor ()                          //获得水果颜色
    {
        return $this->color;
    }
    public function getShape ()                          //获得水果形状
    {
        return $this->shape;
    }
}
```

（2）定义苹果类 Apple，使该类继承自水果类 Fruit，并定义苹果类的构造方法，在构造方法中使用 parent 关键字调用父类构造方法对父类进行初始化，代码如下所示：

```
class Apple extends Fruit                                //苹果类继承水果类
{
    public function __construct ($color, $shape)         //构造方法对类进行初始化
    {
        parent::__construct($color, $shape);
    }
}
```

（3）建立苹果颜色和形状录入表单。用户在表单中输入苹果的颜色和形状后单击“提交”按钮，将通过如下代码打印苹果的颜色和形状属性：

```
if(isset($_POST['color']) && $_POST['color']!=''){                //判断是否已经提交表单
   require 'Fruit.php';                                           //包含水果类
   $apple = new Apple($_POST['color'], $_POST['shape']);          //对水果类进行实例化
   echo '     <font color="blue">我看见的苹果是：'.$apple->getColor().'，'.$apple->getShape().'的</font>';   //打印苹果属性
}
```

上述代码首先判断用户是否已经提交表单，如果是则首先使用 require 语句包含苹果类，然后使用 new 关键字对苹果类进行实例化，最后使用苹果类实例的对象调用父类中的 getColor()方法和 getShape()方法，打印苹果的属性。

秘笈心法

心法领悟 326：使用 final 关键字防止类被继承。

本实例的关键技术中已经说明父类中使用 final 关键字所修饰的方法不能被重写，那么如何使一个类不能被其他类所继承呢？同样可以通过 final 关键字来修饰，如下所示：

```
final class Fruit                                  //使用 final 关键字修饰类
{
}
```

上述的 Fruit 类被 final 关键字所修饰，这样该类就不能被其他类所继承。



实例 327　使用 parent 关键字调用父类的方法　中级

光盘位置：光盘\MR\07\327　趣味指数：★★★

实例说明

本实例主要使用 parent 关键字在小汽车类中调用汽车类的行驶方法。运行本实例，如图 7.17 所示，分别在页面中打印出通过汽车类调用其自身的行驶方法，和其子类小汽车类调用其自身的行驶方法的结果。

行驶

小汽车可以行驶

图 7.17　打印汽车和小汽车的行驶方法

关键技术

在 PHP 5.0 面向对象的编程方式中，使用 parent 关键字加操作符“::”调用父类的方法，调用过程如下所示：

```
class Parent1                                      //定义父类
{
    public function funParent(){                   //定义父类的 funParent()方法

      }
}
class Child extends Parent1                        /定义子类
{
    public function funChild(){                    //定义子类的 funChild()方法
        parent::funParent();                       //调用父类的方法
      }
}
```

使用 parent 关键字不仅可以调用父类的普通方法，而且可以通过调用父类的构造方法实现对父类的初始化。

设计过程

（1）定义汽车类，并在该类中定义汽车的行驶方法，代码如下：

```
class Car                                  //定义汽车类
{
    public function run ()                 //定义行驶方法
    {
        return '行驶';
    }
}
```

（2）定义小汽车类使其继承自汽车类，在类体内定义 smallCarRun()方法，并在该方法中使用 parent 关键字调用汽车类的 run()方法，代码如下：

```
class SmallCar extends Car                 //定义小汽车类，使之继承自汽车类
{
    public function smallCarRun ()         //定义小汽车行驶方法
    {
        return '小汽车可以' . parent::run();
    }
}
```

（3）使用 require 语句包含 Car.php 文件，然后分别对汽车类 Car 和小汽车类 SmallCar 进行实例化，然后分别调用这两个类的行驶方法，代码如下：

```
require 'Car.php';                          //包含汽车类
$car = new Car();                           //对汽车类进行实例化
echo $car->run() . '<br/>';                 //调用汽车类行驶方法
$smallCar = new SmallCar();                 //实例小汽车类
echo $smallCar->smallCarRun();              //调用小汽车类的行驶方法
```

秘笈心法

心法领悟 327：使用 parent 关键字调用父类的静态成员。

在 PHP 5.0 面向对象编程方式中，不仅可以使用 parent 关键字调用父类的普通成员，而且可以使用 parent 关键字加操作符“::”调用父类的静态成员。

实例 328　苹果子类中覆盖水果父类中的方法

光盘位置：光盘\MR\07\328

中级

趣味指数：★★★

实例说明

本实例主要通过在苹果类中重写其父类水果类的 getColor()方法来讲解如何应用类的重写机制。运行本实例，如图 7.18 所示，其中第一行为通过水果类调用其自身的 getColor()方法的输出结果，第二行为苹果类调用其自身的 getColor()方法的输出结果，并且该方法重写了其父类水果类的 getColor()方法。

不同的水果颜色不同，无法确定
苹果的颜色是红色

图 7.18　重写水果类中的 getColor()方法的结果

关键技术

子类不仅可以调用父类的方法，而且可以对指定的方法进行重写，应用重写机制可以扩展类中方法的功能，其中实现对类中方法重写的代码示例如下：

```
class Parent1                               //父类 Parent1
{
    public function fun(){
                                            //父类的 fun()方法
    }
}
class Child extends Parent1                 //子类 Child 继承自 Parent1
{
    public function fun(){
                                            //重写父类的 fun()方法
    }
}
```

对上述代码中的子类 Child 进行实例化，然后使用实例化后的对象调用类中的 run()方法，将执行子类重写后的 run()方法的功能而非父类方法的功能。

设计过程

（1）定义水果类 Fruit，并在该类中定义 getColor()方法，然后定义水果类的子类 Apple，使该类继承自水果类 Fruit，在苹果类 Apple 中重写其父类的 getColor()方法，实现该过程的代码如下所示：

```
class Fruit                                 //定义水果类
{
    public function getColor ()             //获得水果颜色
    {
        return '不同的水果颜色不同，无法确定';
    }
```

```
}
class Apple extends Fruit                   //苹果类继承水果类
{
    private $color;                         //定义苹果颜色属性
    public function __construct ($color)    //构造方法对类进行初始化
    {
        $this->color = $color;
    }
    public function getColor ()             //获得水果颜色
    {
        return '苹果的颜色是' . $this->color;
    }
}
```

（2）包含 Fruit.php 文件，然后使用 new 关键字对 Fruit 类进行实例化，并通过实例化的对象调用该类的 getColor()方法，最后对 Apple 类进行实例化，并调用 Apple 类中的 getColor()方法打印苹果的颜色，该过程的实现代码如下所示：

```
require 'Fruit.php';                        //包含 Fruit.php 文件
$fruit = new Fruit();                       //对水果类进行实例化
echo $fruit->getColor().'<br/>';            //打印水果类的获得颜色方法
$apple = new Apple('红色');                  //对苹果类进行实例化
echo $apple->getColor();                    //打印苹果类的获得颜色方法
```

秘笈心法

心法领悟 328：通过子类对象调用父类中被重写前的方法。

通过子类被实例的对象可以使用操作符“->”直接调用父类中被重写的方法，那么如果想通过子类对象调用父类中未被子类所重写的方法该如何实现呢？可以在子类中定义一个私有变量，该变量用于保存父类对象，然后在子类构造方法中对父类进行实例化，并将实例化后的对象赋给该私有变量，同时在类体中定义一个 getXxx()方法，该方法用于返回父类被实例的对象，这样在类体外就可以通过子类被实例的对象调用 getXxx()方法，然后再使用调用该方法返回的结果调用父类中被重写前的方法即可。实现该过程的示例代码如下：

```
class Parent1                               //定义父类
{
    public function fun ()                  //定义父类的 fun()方法
    {
        return 'Parent';
    }
}
class Child extends Parent1                 //定义子类 Child
{
    private $parent;                        //定义私有成员，用于保存父类对象
    public function __construct ()          //子类构造方法
    {
        $this->parent = new Parent1();      //对父类进行初始化
    }
    public function getParent ()            //返回父类对象
    {
        return $this->parent;
    }
    public function fun ()                  //重写父类的 run()方法
    {
        return 'Child';
    }
}
$child = new Child();                       //对 Child 类进行实例化
echo $child->getParent()->fun();            //以链状方式调用父类未被重写的 run()方法
```

7.4 抽象类和接口

抽象类是为规范面向对象编程提供一种方式，不能被实例化，只能被其他类所继承。接口是为了实现一种

特定功能而预留的类似类的一种类型，只允许定义常量和方法，并且这里的方法没有任何功能实现。本节将通过具体实例讲解 PHP5 面向对象编程方式中，抽象类和接口的应用。

实例 329　美食抽象类　　中级

光盘位置：光盘\MR\07\329　　趣味指数：★★★

实例说明

通过本实例可以演示抽象类不能被实例化，只能被继承的特点。本实例首先定义抽象的父类美食类，然后定义该类的子类面包类，最后分别对这两个类进行实例化并调用获取制作原料的方法。运行本实例，将在页面打印如图 7.19 所示的错误提示，从错误提示可知，抽象类不能被实例化。

如果将抽象类实例的代码注释掉，再次运行本实例，将出现如图 7.20 所示的结果，程序可以正常运行说明抽象类可以被继承。

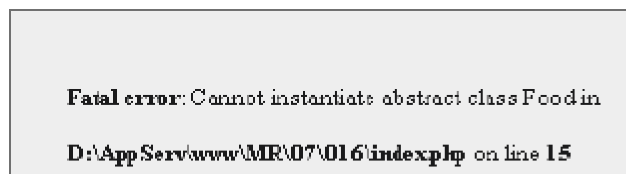


图 7.19　实例抽象类的错误提示

制作面包的材料是：面粉

图 7.20　调用抽象类的子类的结果

关键技术

抽象类与普通类的区别在于，抽象类不能被实例化，即不能用来创建对象，只能被继承。抽象类的定义方法是在 class 前加关键字 abstract，如下所示：

```
abstract class MyClass{
        //类中的属性和方法
}
```

如果对抽象类进行实例化，将在页面打印如图 7.19 所示的错误信息。

设计过程

（1）定义抽象美食类 Food，在该类中定义$material 属性，并通过构造方法对该属性进行初始化，然后定义 getMaterial()方法返回美食的烹制材料，代码如下所示：

```
abstract class Food                                      //美食类
{
    private $material;                                   //烹制材料
    public function __construct ($material)              //父类构造方法
    {
        $this->material = $material;
    }
    public function getMaterial ()                       //获得美食材料方法
    {
        return $this->material;
    }
}
```

（2）定义面包类 Bread，并使其继承自美食类 Food，然后在该类的构造方法中使用 parent 关键字调用父类的构造方法实现对其父类的初始化，该过程的实现代码如下所示：

```
class Bread extends Food                                 //面包类继承美食类
{
    public function __construct ($material)              //子类构造方法
    {
        parent::__construct($material);
    }
}
```

（3）使用 require 语句在 index.php 中包含 Food.php 文件，然后使用 new 关键字对 Bread 类进行初始化，最后通过其实例的对象调用 getMaterial()方法打印出制作面包所需的材料，代码如下所示：

```
require 'Food.php';
$material = '面粉';
//$food = new Food($material);
//echo $food->getMaterial().'<br/>';
$bread = new Bread($material);
echo '制作面包的材料是：'.$bread->getMaterial().'<br/>';
```

秘笈心法

心法领悟 329：PHP 面向对象编程方式只能实现单继承。

上文对 PHP 的继承进行了介绍，但都是单一继承，那么 PHP 能否实现多继承呢？可以通过定义如下类的相互关系进行判断：

```
abstract class A{                           //定义抽象类 A

}
abstract class B{                           //定义抽象类 B

}
class C extends A, B{                       //定义类 C，使其分别继承自 A 和 B

}
```

上述代码中，首先定义抽象类 A 和抽象类 B，然后定义类 C 使其同时继续类 A 和类 B。运行上述代码，将在页面打印如下错误信息，说明 PHP 目前是不能实现多继承的：

Parse error: syntax error, unexpected ',', expecting '{' in D:\AppServ\www\MR\07\016\t.php on line 8

实例 330 学生类多重接口的实现

光盘位置：光盘\MR\07\330

中级

趣味指数：★★★

实例说明

本实例主要介绍 PHP 面向对象编程中接口的定义、应用，以及类如何实现多个接口。运行本实例，如图 7.21 所示，首先在表单中输入学生的学号和学生姓名，然后单击“提交”按钮即可在页面中打印出学生信息。

图 7.21　打印学生信息

关键技术

本实例的关键技术是如何定义接口，以及如何实现一个或多个接口。在 PHP 中接口使用关键字 interface 声明，在类体内不能有属性定义，只能有方法声明，并且不能在方法中定义任何功能代码，PHP 中接口的定义方法如下所示：

```
interface name
{
    public function fun1();
    private function fun2($param);
}
```

对接口定义完成后，可以在定义类时使用关键字 implements 来实现一个或多个接口，所实现的多个接口名称之间使用逗号分割，例如：

```
class MyClass implements interface1, interface2, interface3
{
        //需要实现各个接口中所声明的方法
}
```

综上可知，实现接口的类中至少应包括所实现接口中声明的方法。

设计过程

（1）定义学生 ID 接口，在接口中声明用于设置学生 ID 的 setId()方法和获得学生 ID 的 getId()方法，代码如下所示：

```
interface Property_Id                                   //编号接口
{
    public function setId ($id);                        //方法声明
    public function getId ();
}
```

（2）定义学生名称接口 Property_Name，并在接口中声明设置学生姓名的 setName()方法和获得学生姓名的 getName()方法，实现代码如下所示：

```
interface Property_Name                                 //名称接口
{
    public function setName ($name);                    //方法声明
    public function getName ();
}
```

（3）定义学生类 Student，使其同时实现 Property_Id 接口和 Property_Name 接口，并实现类中所声明方法的定义，代码如下：

```
class Student implements Property_Id, Property_Name     //定义学生类，使其同时实现 Property_Id 和 Property_Name 两个接口
{
    private $id;                                        //编号属性
    private $name;                                      //名称属性
    public function setId ($id)                         //实现接口中的各个方法
    {
        $this->id = $id;
    }
    public function getId ()
    {
        return $this->id;
    }
    public function setName ($name)
    {
        $this->name = $name;
    }
    public function getName ()
    {
        return $this->name;
    }
}
```

（4）建立学生属性信息录入表单，当单击表单中的“提交”按钮后，将在页面中打印出所录入的学生编号和学生名称，代码如下：

```
if(isset($_POST['id']) && $_POST['id']!=''){            //判断是否提交了表单
    require 'Student.php';                              //包含学生类
    $student = new Student();                           //对学生类进行实例化
    $student->setId($_POST['id']);                      //设定学生 ID
    $student->setName($_POST['name']);                  //设定学生名称
    echo '<font color="green">当前学生的学号是：'.$student->getId().'  姓名是：'.$student->getName().'</font>';      //打印学生信息
}
```

上述代码中，首先判断是否已经提交了表单，如果是，则首先使用 require 语句包含 Student.php 文件，然后使用 new 关键字对学生类 Student 进行实例化，并调用实例化后对象的 setXxx()方法分别为学生编号和学生姓名赋初值，最后分别调用类中的 getXxx()方法打印学生信息。

秘笈心法

心法领悟 330：require 语句与 include 语句的区别。

require 语句与 include 语句的主要区别有如下两点：

☑ 只要程序运行就会将 require 语句所包含的文件包含进来，而 include 语句则是执行到该语句时包含指定的文件。

☑ 如果包含的语句不存在，require 语句将给出错误提示，而 include 语句给出警告。

7.5 类的多态

多态是面向编程方式的重要特点之一，通过多态可以为类提供功能上的多种实现，PHP 中可以通过继承或接口来实现类的多态。

实例 331 通过继承实现多态

光盘位置：光盘\MR\07\331

中级

趣味指数：★★★

实例说明

本实例主要通过继承表现类的多态。在制作本实例时，首先定义动物类 Animal，然后分别定义企鹅类 Penguin 和昆虫类 Insect，并在这两个子类中分别重写动物类的行走方法，最后通过这两个子类实例的对象调用其自身的行走方法，将在页面打印如图 7.22 所示的内容。

通过继承实现多态：
企鹅可以直立行走
昆虫可以爬行

图 7.22 通过继承实现类的多态

关键技术

本实例的关键是如何通过继承来实现类的多态。在 PHP 面向对象编程中，可以首先定义一个抽象的父类，然后再定义多个子类来继承该父类，在子类中可以通过方法重载的方式重写父类中的功能，这样就实现了类的多态，代码如下所示：

```
abstract class Parent                          //定义抽象的父类 Parent
{
    public function fun ()                     //定义父类的 fun()方法
    {
        return '父类 fun()方法的实现';
    }
}
class Child1 extends Parent                    //定义子类 Child1，使其继承 Parent
{
    public function fun ()                     //在 Child1 中重写 Parent 类中的 fun()方法
    {
        return 'Child1 中 fun()方法的实现';
    }
}
class Child2 extends Parent                    //定义子类 Child2， 使其继承 Parent
{
    public function fun ()                     //在 Child2 中重写 Parent 类中的 fun()方法
    {
        return 'Child2 中 fun()方法的实现';
    }
}
```

设计过程

（1）定义抽象的动物类 Animal，并在该类中定义行走的方法 walk()，然后定义企鹅 Penguin 和昆虫类 Insect，使二者分别继承类 Animal，并在这两个类中重写 Animal 类中的 walk()方法，代码如下所示：

```
abstract class Animal                          //定义动物类
{
    public function walk ()                    //定义动物类中的行走方法
    {
        return '动物能行走';
    }
```

```
}
class Penguin extends Animal                    //定义企鹅类，并使其继承自动物类
{
    public function walk ()                     //重写父类中的 walk()方法
    {
        return '企鹅可以直立行走';
    }
}
class Insect extends Animal                     //定义昆虫类，并使其继承自动物类
{
    public function walk ()                     //重写父类中的 walk()方法
    {
        return '昆虫可以爬行';
    }
}
```

（2）使用 require 语句包含 Animal.php 文件，然后分别对企鹅类和昆虫类进行实例化，并分别调用这两个类中的 walk()方法，代码如下所示：

```
require 'Animal.php';                           //包含 Animal.php 文件
$penguin = new Penguin();                       //对企鹅类进行实例化
echo $penguin->walk();                          //调用企鹅类的 walk()方法
echo '<br/>';
$insect = new Insect();                         //对昆虫类进行实例化
echo $insect->walk();                           //调用昆虫类的 walk()方法
```

通过上述实例的运行结果可知，企鹅类和昆虫类虽然都重写了父类的 walk()方法，但在页面中的打印结果却不同，这就是通过继承实现类多态的体现。

秘笈心法

心法领悟 331：使用 abstract 关键字防止类被实例化。

在 PHP 面向对象编程中，如果所定义的类只能被其他类继承，而自身并不需要进行实例化，可以使用 abstract 关键字将该类声明为抽象类。

例如定义抽象类 A，代码如下所示：

```
<?php
abstract class A{                               //定义抽象类 A

}
new A();                                        //对类 A 进行实例化
```

如果对类 A 使用 new 关键字进行实例化，将在页面中打印如下错误信息，从而说明抽象类是不能被实例化的。

```
Fatal error: Cannot instantiate abstract class A in D:\AppServ\www\MR\07\018\t.php on line 5
```

实例 332　通过接口实现多态

光盘位置：光盘\MR\07\332

中级　趣味指数：★★★

实例说明

本实例主要介绍如何通过接口来表现类的多态性。制作本实例时，首先定义动物接口 Animal，然后分别定义企鹅类和昆虫类来实现这个接口，并实现接口中所声明的 walk()方法，运行本实例，如图 7.23 所示。

通过接口实现多态：
企鹅可以直立行走
昆虫可以爬行

图 7.23　通过接口实现类的多态

关键技术

在继承中通过方法重写方式可以体现类的多态性，通过接口的方式同样可以体现类的多态性，代码如下所示：

```
interface A                                     //定义接口 A
```

```
{
    public function fun ();                         //声明接口中的 fun()方法
}
class Class1 implements A                           //定义类 Class1 使其实现接口 A
{
    public function fun ()                          //实现接口 A 中的 fun()方法
    {
        return '在类 Class1 中实现接口 A 中的 fun()方法';
    }
}
class Class2 implements A
{
    public function fun ()                          //实现接口 A 中的 fun()方法
    {
        return '在类 Class2 中实现接口 A 中的 fun()方法';
    }
}
```

上述代码中，定义接口 A，并在接口中声明 fun()方法，然后分别定义类 Class1 和类 Class2，使这两个类都实现接口 A，并实现接口 A 中的 fun()方法。从示例代码中可知，类 Class1 和类 Class2 虽然都实现了接口 A 中所声明的 fun()方法，但这两个方法实现的功能却不同，从而体现类的多态性。

设计过程

（1）定义动物类接口 Animal，并在该接口中声明 walk()方法，然后定义企鹅类 Penguin 和昆虫类 Insect，使这两个类分别实现 Animal 接口，代码如下所示：

```
interface Animal                                    //定义动物接口
{
    public function walk ();                        //声明行走的方法
}
class Penguin implements Animal                     //定义企鹅类，并使其实现动物接口
{
    public function walk ()                         //重写父类中的 walk()方法
    {
        return '企鹅可以直立行走';
    }
}
class Insect implements Animal                      //定义昆虫类，并使其实现动物接口
{
    public function walk ()                         //重写父类中的 walk()方法
    {
        return '昆虫可以爬行';
    }
}
```

（2）分别对企鹅类和昆虫类进行实例化，然后使用实例化的对象调用类中的 walk()方法，代码如下所示：

```
require 'Animal.php';                               //包含 Animal.php 文件
$penguin = new Penguin();                           //对企鹅类进行实例化
echo $penguin->walk();                              //调用企鹅类的 walk()方法
echo '<br/>';
$insect = new Insect();                             //对昆虫类进行实例化
echo $insect->walk();                               //调用昆虫类的 walk()方法
```

秘笈心法

心法领悟 332：抽象类和接口的区别。

☑ 在定义上，在抽象类中可以对方法所实现的功能进行具体定义，而在接口中只能对方法进行声明，不能具体实现方法的功能。

☑ 在用法上，继承抽象类的子类可以重写父类的方法，或通过实例后的对象直接调用父类中的方法，而实现接口的类中，必须包含所实现接口的所有方法。

7.6　常用关键字

在 PHP 面向对象编程中，有些关键字是需要掌握的，如 final、static、clone、instanceof 等，本节将通过具体实例讲解这些关键字的使用方法。

实例 333	使用 final 关键字防止类被继承 光盘位置：光盘\MR\07\333	中级 趣味指数：★★★

实例说明

通过本实例主要讲解 final 关键字的用法。首先定义水果类 Fruit，然后定义 final 型的苹果类 Apple，使之继承自水果类，通过 Apple 类即可实现对苹果属性的设置。运行本实例如图 7.24 所示，首先在文本框中输入苹果的颜色和形状属性，然后单击“提交”按钮，即可在页面中打印出苹果的属性信息。如果再定义一个类，并使之继承 final 型的 Apple 类，则会在页面中打印出如图 7.25 所示的错误提示信息。

苹果的颜色是：
红色
苹果的形状是：
圆形　提交
苹果是红色圆形的

图 7.24　打印苹果的属性

苹果的颜色是：
苹果的形状是：
提交
Fatal error: Class Test may not inherit from final class (Apple) in **D:\AppServ\www\MR\07\020\Fruit.php** on line **30**

图 7.25　继承 final 型水果类的错误提示

关键技术

通过本实例的运行结果可知，使用 final 关键字修饰的类只能被实例化，不能被继承，如果继承了 final 型的类，将在页面中打印如下错误提示：

Fatal error: Class Test may not inherit from final class (Apple) in D:\AppServ\www\MR\07\020\Fruit.php on line 30

设计过程

（1）定义水果类 Fruit，在该类中定义水果的颜色属性$color，并通过构造方法对颜色属性进行初始化，然后定义获得颜色的 getColor()方法，代码如下所示：

```
class Fruit                                    //定义水果类
{
    private $color;                            //定义颜色属性
    public function __construct ($color)       //通过构造方法对颜色属性进行初始化
    {
        $this->color = $color;
    }
    public function getColor ()                //获得水果颜色的方法
    {
        return $this->color;
    }
}
```

（2）定义 final 型的苹果类 Apple，并使其继承水果类 Fruit，在水果类中定义苹果的形状属性$shape，通过构造方法对苹果类进行初始化，最后定义 getShape()方法返回水果的形状，代码如下所示：

```
final class Apple extends Fruit                //定义 final 型的苹果类，使之继承水果类
```

```
{
    private $shape;                                    //定义形状属性
    public function __construct ($color, $shape)       //构造函数
    {
        parent::__construct($color);                   //调用父类构造方法
        $this->shape = $shape;                         //对形状属性进行初始化
    }
    public function getShape ()                        //获得苹果的形状方法
    {
        return $this->shape;
    }
}
/*
class Test extends Apple {                             //定义一个 Test 测试类，使之继承 final 形的 Apple 类

}
*/
```

上述代码中，以多行注释的部分用于测试被关键字 final 修饰的类是否可以被继承。

（2）建立苹果属性录入表单，当用户单击表单中的“提交”按钮后，将通过如下代码打印出苹果的属性：

```
if(isset($_POST['color']) && $_POST['color']!=''){
    require 'Fruit.php';
    $apple = new Apple($_POST['color'], $_POST['shape']);
    echo '<font color="red">苹果是'.$apple->getColor().$apple->getShape().'的</font>';
}
```

上述代码中，首先判断用户是否已经提交了表单，如果是，则首先使用 require 语句包含 Fruit.php 文件，然后对苹果类进行实例化，最后分别通过 getColor()方法和 getShape()方法打印出苹果的属性。

秘笈心法

心法领悟 333：使用函数 isset()判断用户是否提交了表单。

在实际项目开发中，经常需要将表单提交到当前页面进行处理，而处理表单所提交内容的代码需要在表单提交后执行，那么如何判断用户已经提交了表单呢？在 PHP 中可以通过函数 isset()实现，该函数的语法格式如下所示：

```
bool isset ( mixed var [, mixed var [, ...]] )
```

该函数的参数为一个变量列表，如果所指定的全部变量都已经设置值则返回 TRUE，反之有一个没有被设置值则返回 FALSE，例如本实例中通过如下代码判断是否已经提交了表单：

```
if(isset($_POST['color']) && $_POST['color']!=''){          //判断用户是否已经提交了表单
                                                            //提交表单后的处理代码
}
```

实例 334　使用 static 关键字定义类的静态成员

光盘位置：光盘\MR\07\334

中级　趣味指数：★★★

实例说明

本实例主要讲解 PHP 面向对象编程方式中的 static 关键字的用法，运行本实例，如图 7.26 所示，首先在图中的文本框中输入要计算的数字，同时在计算类型列表框中选择计算类型，然后单击“求值”按钮，即可在页面中打印出计算结果。这里要注意，在进行除运算时，如果除数为 0，则应该给出错误提示，如图 7.27 所示。

数值计算器：
12
/
3　求值
结果：4

图 7.26　数值计算器

数值计算器：
6
/
0　求值
结果：除数不能为0

图 7.27　除数为 0 时的提示信息

关键技术

在 PHP 面向对象编程方式中，使用 static 关键字修饰的属性称为静态属性，使用 static 关键字修饰的方法称为静态方法，通过类实例的对象可以调用类中的公有方法，而静态成员的特点是在不对类进行实例化的前提下，直接通过类名即可对其进行调用，其调用格式如下所示：

```
类名::公有静态成员
```

设计过程

（1）定义数值计算类 Math，并在 Math 类中定义用于进行加、减、乘、除运算的静态方法，在进行求除运算中通过 try/catch 语句进行除数为 0 时的异常处理，实现该过程的代码如下所示：

```
class Math                                              //定义数值计算类
{
    public static function add ($num1, $num2)           //相加的方法
    {
        return $num1 + $num2;
    }
    public static function sub ($num1, $num2)           //相减的方法
    {
        return $num1 - $num2;
    }
    public static function multi ($num1, $num2)         //相乘的方法
    {
        return $num1 * $num2;
    }
    public static function div ($num1, $num2)           //相除的方法
    {
        try {
            if ($num2 == 0) {                           //如果除数为 0，则抛出异常
                throw new Exception('除数不能为 0');
            } else {
                return $num1 / $num2;
            }
        } catch (Exception $e) {
            return $e->getMessage();                    //如果除数为 0，则给出错误提示
        }
    }
}
```

（2）定义数值录入表单，当在表单中录入数字，并单击表单的“求值”按钮后，将通过如下代码计算出表单中所定义的算数运算的结果：

```
if (isset($_POST['num1']) && trim($_POST['num1']) != '') {   //判断是否提交了表单
    require 'Math.php';                                      //包含 Math.php 文件
    $num1 = $_POST['num1'];                                  //获得提交的数字
    $num2 = $_POST['num2'];
    switch ($_POST['type']) {
        case '+':                                            //进行加运算
            $result = Math::add($num1, $num2);
            break;
        case '-':                                            //进行减运算
            $result = Math::sub($num1, $num2);
            break;
        case '*':                                            //进行乘运算
            $result = Math::multi($num1, $num2);
            break;
        case '/':                                            //进行除运算
            $result = Math::div($num1, $num2);
            break;
    }
    echo '结果：' . $result;                                  //打印计算结果
}
```

上述代码中，首先判断表单是否已经提交，如果是，则使用 switch 语句判断要进行计算的类型，然后通过 Math 类直接调用其中的静态方法进行计算。

秘笈心法

心法领悟 334：直接通过类名调用类中的静态方法时不调用构造方法和析构方法。

如果直接通过类名调用类中的静态方法，而不通过 new 关键字对类进行实例化，是否也会执行类中的构造方法和析构方法呢？下面定义如下代码，在下述代码中，定义类 Test，然后分别在类中定义构造方法、析构方法和静态方法 fun()，最后使用类名直接调用静态方法 fun()。

```
class Test                                          //定义 Test 类
{
    public function __construct ()
    {                                               //构造方法
        echo '执行构造方法<br/>';
    }
    static public function fun ()
    {                                               //静态方法
        echo '执行静态方法';
    }
    public function __destruct ()
    {                                               //析构方法
        echo '执行析构方法';
    }
}
Test::fun();                                        //通过类名调用类中静态方法
```

运行上述代码后，将在页面中打印“执行静态方法”，从运行结果可以得知，如果通过类名调用类中的静态方法，不调用构造方法和析构方法。

实例 335 使用 clone 关键字实现对象的克隆

光盘位置：光盘\MR\07\335

中级

趣味指数：★★★

实例说明

本实例主要讲解 PHP 5.0 面向对象编程中 clone 关键字的使用方法，运行本实例，分别如图 7.28 和图 7.29 所示，其中图 7.28 中的输出结果是使用 clone 关键字克隆对象后，分别用原来对象和克隆的对象对类中方法进行重新赋值并调用的结果，而图 7.29 为使用等号赋值的方式来产生一个新对象后，再分别使用新旧对象实现对类中的方法进行调用的结果。

羊的颜色是白色
羊的颜色是灰色

图 7.28　使用 clone 关键字传递对象

羊的颜色是白色
羊的颜色是白色

图 7.29　使用等号传递对象

关键技术

在 PHP 5.0 以后的版本中，使用 clone 关键字实现对对象的克隆，该关键字的语法格式如下：

```
$obj_new = clone $obj_old;
```

初学者可能会问，直接用等号就可以将当前对象赋给一个其他对象，为什么还要用 clone 关键字？这主要是因为，在 PHP 5.0 中，对象被存储于独立的结构 Object Store 中，而不像其他一般变量存储于 Zval 中。在 Zval 中仅存储对象地址的引用，当赋值一个变量或者将一个变量传递给一个函数时，就不再复制数据了，所以对新变量的更改不会影响原变量的内容，而采用 Object Store 存储对象时，使用等号产生一个新对象后，对新对象的

更改会导致原对象的更改，而使用 clone 关键字克隆对象，可避免这一问题。

说明：使用 clone 关键字克隆一个对象，就好比克隆一个人，打一下新克隆出的人，被克隆的人并不会感到疼痛，其中的道理是一样的。

设计过程

（1）建立羊类 Sheep，在该类中定义羊的颜色属性$color，同时定义用来设置颜色的 setColor()方法和获得颜色的 getColor()方法，代码如下所示：

```
class Sheep
{
    private $color;                          //颜色属性
    public function setColor ($color)        //设置颜色的方法
    {
        $this->color = $color;
    }
    public function getColor ()              //获得颜色的方法
    {
        return '羊的颜色是' . $this->color;
    }
}
```

（2）对 Sheep 类进行实例化，然后分别用等号赋值和用 clone 方法测试对对象进行更改的影响结果，代码如下所示：

```
require 'Sheep.php';                  //包含 Sheep.php 文件
$sheep = new Sheep();                 //对 Sheep 类进行实例化
/*
$sheep->setColor('白色');             //设置羊的颜色
echo $sheep->getColor();              //打印羊的颜色
$sheep1 = $sheep;                     //将$sheep 对象赋值给新对象$sheep1
echo '<br/>';
$sheep1->setColor('灰色');            //通过新对象$sheep1 调用设置颜色的方法
echo $sheep->getColor();              //打印羊的颜色
*/

$sheep->setColor('白色');             //设置羊的颜色
echo $sheep->getColor();              //打印羊的颜色
$sheep1 = clone $sheep;               //克隆$sheep 对象，产生一个新的$sheep1 对象
echo '<br/>';
$sheep1->setColor('灰色');            //通过新克隆的$sheep1 对象调用 Sheep 类中的 setColor()方法
echo $sheep->getColor();              //打印羊的颜色
```

秘笈心法

心法领悟 335：克隆对象的比较方法。

在 PHP 中可以使用“==”和“===”操作符实现对对象的比较，为了明确这两种操作符的差别，首先看下面的例子。

```
class Test                            //定义 Test 类
{
    //类体
}
$test = new Test();                   //对 Test 类进实例化
$test1 = clone $test;                 //克隆一个新的$test1 对象
echo intval(($test == $test1)) . '<br/>' . intval(($test === $test1));    //打印比较结果
```

运行上述代码，将在页面中打印 1 和 0。原因就是在对象比较时，“==”比较两个对象的内容，而“===”则比较两个对象的句柄，所以“$test==$test1”比较的是两个对象的内容，返回结果应该为 TRUE，而“$test===$test1”则比较两个对象的句柄，即引用地址，所以返回 FALSE。

实例 336 检测当前对象属于哪个类

光盘位置：光盘\MR\07\336

中级

趣味指数：★★★

实例说明

本实例主要讲解 PHP 面向对象编程中，instanceof 关键字的使用方法。运行本实例，将在页面中打印出如图 7.30 所示的结果。本实例主要定义了苹果类和桔子类，然后分别对这两个类进行实例化，同时定义一个根据传递的对象自动识别对象所属类的方法。

苹果是：红色
桔子是：橙色

图 7.30　根据不同的对象调用类中的方法

关键技术

instanceof 关键字用于检测某实例是否属于某一个类的类型，如果是，则返回 TRUE，反之返回 FALSE，该关键字的使用格式如下：

```
对象名 instanceof 类名
```

设计过程

（1）定义苹果类 Apple 和桔子类 Orange，然后分别在这两个类中定义获得颜色的方法 getColor()，代码如下所示：

```
class Apple                                  //定义苹果类
{
    public function getColor ()              //获得苹果颜色的方法
    {
        return '红色';
    }
}
class Orange                                 //定义桔子类
{
    public function getColor ()              //获得桔子颜色的方法
    {
        return '橙色';
    }
}
```

（2）使用 require 语句包含 Fruit.php 文件，然后定义根据对象打印水果颜色的方法，最后使用 new 关键字分别对 Apple 类和 Orange 类进行实例化，并使用 getColor()方法打印出相应水果的颜色，代码如下所示：

```
require 'Fruit.php';                         //包含水果类
function getColor ($obj)                     //定义根据对象类型获得相应水果颜色的方法
{
    if ($obj instanceof Apple) {             //判断是否为苹果实例的对象
        $str = '苹果是：';
    } elseif ($obj instanceof Orange) {      //判断是否为桔子实例的对象
        $str = '桔子是：';
    }
    return $str . $obj->getColor();          //打印结果
}
$apple = new Apple();                        //对苹果类进行实例化
$orange = new Orange();                      //对桔子类进行实例化
echo getColor($apple) . '<br/>';             //打印苹果颜色
echo getColor($orange);                      //打印桔子颜色
```

秘笈心法

心法领悟 336：使用函数 is_a()判断指定的对象是否属于某类或其子类的对象。

从本实例可知，使用 instanceof 关键字可以检测指定的对象是否属于某一个类的类型，而 PHP 中又提供了

函数 is_a()，该函数用于判断指定的对象是否属于某类或其子类的对象，其语法格式如下：

```
bool is_a ( object object, string class_name )
```

如果对象 object 是 class_name 所代表类或父类的对象则返回 TRUE，否则返回 FALSE。

7.7　魔 术 方 法

PHP 5.0 中，主要提供了__set()、__get()、__isset()、__unset()、__clone()、__toString()和__autoload()等魔术方法，本节将通过具体实例讲解 PHP 中魔术方法的含义和具体使用方法。

实例 337　使用__set()方法为类中未声明的属性赋值

中级

光盘位置：光盘\MR\07\337

趣味指数：★★★

实例说明

本实例主要讲解如何使用 PHP 中的__set()方法存取类中未声明的属性。运行本实例如图 7.31 所示，在页面中以表格的形式列出各本图书的详细信息，其中图书的“备注”属性使用__set()方法赋值。

书　名	页　码	作　者	价　格	备　注
《PHP从基础到**》	650	小张、小潘、小王	58	备注
《PHP函数**》	800	小潘、小王	80	备注
《PHP范例**》	700	小李、小懂	85	备注
《PHP实战**》	750	小郭、小刘	75	备注

图 7.31　图书信息列表

关键技术

魔术方法__set()的作用是存取类中未声明的属性，该方法必须接收两个参数，分别用来表示类中未声明的属性名和属性值，__set()方法的语法格式如下：

```
function __set($name, $value)
{
        //可以使用$name 和$value 在方法中存取类中未定义的属性名和数值
}
```

上述__set()方法中的参数$name 表示类中未声明属性的名称，$value 表示类中未声明属性的值。

设计过程

（1）定义图书类 Book，在类中定义图书的名称、页码、作者和价格等属性，并分别定义这些属性的 setXxx()和 getXxx()方法，然后在类中声明__set()方法，使用该方法为类中未声明的属性赋初值，实现该过程的代码如下：

```
class Book                                   //定义图书类
{
    private $name;                           //书名
    private $page;                           //页码
    private $writer;                         //作者
    private $price;                          //价格
    private $other;                          //其他信息
    public function setName ($name)          //设置书名
    {
        $this->name = $name;
    }
    public function getName ()               //获得书名
    {
        return $this->name;
    }
```

```
    public function setPage ($page)                    //设置页码
    {
        $this->page = $page;
    }
    public function getPage ()                         //获得页码
    {
        return $this->page;
    }
    public function setWriter ($writer)                //设置作者
    {
        $this->writer = $writer;
    }
    public function getWriter ()                       //获得作者
    {
        return $this->writer;
    }
    public function setPrice ($price)                  //设置价格
    {
        $this->price = $price;
    }
    public function getPrice ()                        //获得价格
    {
        return $this->price;
    }
    public function __set ($name, $value)
    {
        $this->other = $value;
    }
    public function getOther ()
    {
        return $this->other;
    }
}
```

（2）建立图书信息二维数组，用于保存图书信息，代码如下所示：

```
require 'Book.php';
$arrayBook = array(                                    //图书信息数组，用于模拟图书信息数据表
    array('name'=>'《PHP 从基础到**》', 'page'=>'650', 'writer'=>'小张、小潘、小王','price'=>'58'),
    array('name'=>'《PHP 函数**》', 'page'=>'800', 'writer'=>'小潘、小王','price'=>'80'),
    array('name'=>'《PHP 范例**》', 'page'=>'700', 'writer'=>'小李、小懂','price'=>'85'),
    array('name'=>'《PHP 实战**》', 'page'=>'750', 'writer'=>'小郭、小刘','price'=>'75')
);
```

（3）使用 foreach 语句通过循环输出图书信息。在 foreach 循环体内部，首先对 Book 类进行实例化，然后分别调用类中的 setXxx()方法为类中已声明的属性赋初值，同时为类中的未声明的备注属性赋值，最后使用 getXxx()方法打印出图书的信息，实现该过程的代码如下所示：

```
<?php
require 'db.php';
foreach ($arrayBook as $key => $aBook) {
    $book = new Book();
    $book->setName($aBook['name']);
    $book->setPage($aBook['page']);
    $book->setWriter($aBook['writer']);
    $book->setPrice($aBook['price']);
    $book->bz = '备注';
?>
<div style="width:100%; <?php if($key < count($arrayBook)-1){?>border-bottom:1px solid #0463BD;<?php } ?> clear:both;">
    <div style="width:160px; height:22px; line-height:22px; text-align:left; float:left;">
      <?php echo $book->getName()?>
    </div>
    <div style="width:160px; height:22px; line-height:22px; border-left:1px solid #0463BD; float:left;">
    <?php echo $book->getPage()?>
    </div>
    <div style="width:160px; height:22px; line-height:22px; border-left:1px solid #0463BD; float:left;">
    <?php echo $book->getWriter()?>
    </div>
    <div style="width:160px; height:22px; line-height:22px; border-left:1px solid #0463BD; float:left;">
    <?php echo $book->getPrice()?>
```

```
        </div>
        <div  style="width:156px; height:22px; line-height:22px; border-left:1px solid #0463BD; float:left;">
        <?php echo $book->getOther()?>
        </div>
</div>
<?php }?>
```

秘笈心法

心法领悟 337：在类体外直接为类中未定义的属性赋初值。

在 PHP 面向对象编程中，在类体外可以直接为类中未定义的属性赋初值，其示例代码如下所示：

```
class Test                                              //定义类 Test
{

}
$test = new Test();                                     //对 Test 类进行实例化
$test->x = '明日';                                       //为 Test 类中未定义的属性 x 赋初值
echo $test->x;                                          //打印 x 属性
```

上述代码中，首先定义类 Test，然后再使用 new 关键字对 Test 类进行实例化，并使用该类实例的对象调用类中未定义的属性 x，同时为该属性赋初值，最后打印该属性。运行上述代码将在页面中打印出“明日”二字，从而说明即使在类中未定义某属性，在类体外也可以对该属性进行操作。

实例 338　使用__get()方法获取未定义属性的名称

光盘位置：光盘\MR\07\338

中级

趣味指数：★★★

实例说明

本实例主要应用__get()方法获得类中未定义属性的名称。运行本实例如图 7.32 所示，页面中打印的内容是通过调用苹果类的属性打印的结果，而弹出的提示对话框是因为没有在类中定义$produceArea 属性而通过__get()方法弹出的。



图 7.32　类中未定义属性提示框

关键技术

魔术方法__get()用于获得类中不存在属性的名称，该方法必须带有一个表示未定义属性名称的参数，__get()方法的使用格式如下所示：

```
function __get($name)
{
        //在方法内部可以使用$name 的值，该值即为未定义属性的名称
}
```

设计过程

（1）定义苹果类 Apple，在该类中分别定义苹果颜色、形状和重量等属性，并使用构造方法对这些属性进行初始化，然后定义用于获得苹果属性信息的 getProperty()方法，最后定义__get()方法用于弹出提示未定义属性名称的对话框，实现该过程的代码如下：

```
class Apple
{
    private $color;                                     //颜色
    private $shape;                                     //形状
    private $weight;                                    //重量
    public function __construct ($color, $shape, $weight)   //构造方法
    {
        $this->color = $color;
        $this->shape = $shape;
```

```
        $this->weight = $weight;
    }
    public function getProperty ()
    {
        return '这个苹果重' . $this->weight . '，是' . $this->color . '，' . $this->shape . '的！';
    }
    public function __get ($name)
    {                                                    //使用__get()方法弹出未定义属性的提示
        echo '<script>alert("在类中未定义属性' . $name . '！");</script>';
    }
}
```

（2）使用 new 关键字对苹果类 Apple 进行实例化，然后使用实例化后的对象调用苹果类中的 getProperty()方法打印苹果的属性信息，最后使用苹果对象调用类中未定义的属性，实现该过程的代码如下所示：

```
require 'Apple.php';                                     //使用 require 语句包含 Apple.php 文件
$apple = new Apple('红色', '圆形', '0.4kg');             //使用 new 关键字对 Apple 类进行实例化
echo $apple->getProperty();                              //调用类中的 getProperty()属性
echo $apple->produceArea;                                //调用类中未定义的属性
```

秘笈心法

心法领悟 338：使用__get()方法实现错误信息的提示。

在 PHP 面向对象编程方式中，为了防止因调用类中未定义的属性而导致程序运行期错误或程序功能的不稳定，可以在类中指定__get()方法实现未定义属性的错误信息提示。

实例 339 使用__call()方法打印类中未定义方法的信息

光盘位置：光盘\MR\07\339

中级

趣味指数：★★★

实例说明

本实例主要讲解__call()魔术方法的使用。运行本实例，如图 7.33 所示，将在页面中打印出图书的信息和调用未定义方法的名称。其中图书的信息是图书类调用类中的 getProperty()方法所返回的结果，而未定义 getInfo 方法的提示是通过类中的__call()方法的打印结果。

《PHP范例**》的价格是85元

getInfo方法未定义

图 7.33 类中未定义属性提示框

关键技术

魔术方法__call()在调用类中一个不存在或不可见的方法时执行，__call()方法必须接收两个参数，用来存放视图调用的方法名及其参数，该方法的语法格式如下：

```
function __call($name, $arguments)
{

}
```

上述示例代码中的参数$name 用于表示类中未定义方法的名称，$arguments 用于表示未定义方法的参数所组成的数组。

设计过程

（1）定义图书类 Book，首先在该类中定义$name 和$price 属性，并使用构造方法对这两个参数进行实例化，然后定义 getProperty()方法返回图书属性，最后定义__call()方法打印未定义方法的名称，实现该过程的代码如下所示：

```
class Book                                               //图书类
{
    private $name;                                       //书名
    private $price;                                      //价格
```

```
    public function __construct ($name, $price)              //构造方法
    {
        $this->name = $name;
        $this->price = $price;
    }
    public function __call ($name, $arguments)               //__call()方法
    {
        echo $name . '方法未定义';
    }
    public function getProperty ()                           //获得图书信息的方法
    {
        return $this->name . '的价格是' . $this->price . '元';
    }
}
```

（2）使用 new 关键字对 Book 类进行实例化，然后使用实例化的对象调用类中的 getProperty()方法打印图书信息，最后使用实例后的对象调用类中未定义的 getInfo()方法，代码如下所示：

```
require 'Book.php';                                  //包含 Book.php 文件
$bookName = '《PHP 范例**》';                         //书名
$price = '85';                                       //价格
$book = new Book($bookName, $price);                 //对 Book 类进行实例化
echo $book->getProperty() . '<br/>';                 //打印图书属性
$book->getInfo($bookName, $price);                   //调用类中未定义的 getInfo()方法
```

秘笈心法

心法领悟 339：在类中未定义__call()魔术方法时，调用类中未定义的方法会提示错误信息。

在类体外可以使用类实例的对象为类中未定义的属性赋值，但是如果类中没有定义__call()方法，直接调用类中未定义的方法时会提示错误信息，通过如下示例代码进行演示：

```
class Test                                           //定义类 Test
{

}
$test = new Test();                                  //对 Test 类进行实例化
$test->fun();                                        //调用类中未定义的方法
```

在上述示例代码中，定义类 Test，并使用 new 关键字对其进行实例化，然后使用类实例化的对象调用类中未定义的 fun()方法，将在页面打印如下错误提示。

Fatal error: Call to undefined method Test::fun() in D:\AppServ\www\MR\07\t.php on line 7

实例 340	使用__toString()方法将类的实例转化为字符串 光盘位置：光盘\MR\07\340	中级 趣味指数：★★★

实例说明

本实例主要讲解如何通过__toString()魔术方法将类实例的对象转换为字符串，运行本实例，如图 7.34 所示，在页面中打印出圆的半径和该半径对应的圆面积，其中圆的面积是直接打印圆类所实例对象的结果。

圆的半径是：2

圆的面积是：12.57

图 7.34　打印圆的半径和面积

关键技术

魔术方法__toString()的作用是将类实例的对象转换为字符串，该方法的语法格式如下：

```
function __toString(){
    return 字符串;
}
```

该方法的返回值必须为字符串类型，如果在类中未定义该方法，当直接输出某一个类实例的对象时，将在页面中打印如下错误提示：

Catchable fatal error: Object of class Test could not be converted to string in D:\AppServ\www\MR\07\027\t.php on line 3

当在类中定义__toString()方法后，直接输出类实例后的对象将在页面中打印__toString()方法所返回的字符串。

设计过程

（1）创建圆类 Circle，在圆类中定义圆半径私有属性$radius，并使用构造方法对该属性进行初始化，然后定义__toString()魔术方法，使用该方法返回圆的面积值，代码如下所示：

```
class Circle                                                    //定义圆类
{
    private $radius;                                            //圆半径属性
    public function __construct ($radius)                       //构造方法
    {
        $this->radius = $radius;
    }
    public function __toString ()                               //定义__toString()方法
    {
        return "圆的面积是：" . (string) number_format((pi() * pow($this->radius, 2)), 2);
    }
}
```

（2）使用 require 语句包含 Circle.php 文件，然后设定圆的半径为 2，并使用 echo 语句输出圆的半径和圆类实例的对象，代码如下所示：

```
require 'Circle.php';                                           //包含 CircleArea.php 文件
$radius = 2;                                                    //圆半径
echo '圆的半径是：'.$radius.'<br/>';                              //打印圆半径的值
echo new Circle($radius);                                       //打印圆面积
```

秘笈心法

心法领悟 340：使用函数 number_format()对数字进行格式化输出。

在对数字、货币等数值型数据进行操作时，经常需要对数字进行格式化输出，PHP 中提供了专门对数字进行格式化输出的函数 number_format()，其语法格式如下：

```
string number_format ( float number [, int decimals [, string dec_point, string thousands_sep]] )
```

参数说明：

number：必要参数，指定要进行格式化的字符串。

decimals：可省参数，指定要保留小数的位数，如果省略该参数则保留 3 位小数。

dec_point：可省参数，用该参数指定的字符替代小数点，该参数与参数 thousands_sep 同时使用。

thousands_sep：可省参数，用该参数指定的字符替代千位分隔符，该参数与参数 dec_point 同时使用。

实例 341 使用__isset()方法提示未定义属性信息

光盘位置：光盘\MR\07\341

中级

趣味指数：★★★

实例说明

运行本实例，如图 7.35 所示，在页面中打印出未定义属性的提示信息以及“小明”的年龄和体重信息，其中类中未定义属性 sex 的提示信息是通过魔术方法__isset()打印的。

在类中未定义属性sex

小明的年龄是：12岁，体重是：45公斤

图 7.35　打印人的属性信息

关键技术

魔术方法__isset()是当使用 isset()函数判断类中未定义的属性是否被设置时所调用的，该方法的使用格式如下所示：

```
function __isset ($name)
{

}
```

魔术方法__isset()必须包含一个用于代表类中未定义属性的名称参数。

设计过程

（1）定义人类 Person，在该类中定义人的名称、年龄和体重等属性，并定义与这些属性相对应的 setXxx()方法和 getXxx()方法，代码如下所示：

```
class Person
{
    private $name;                                  //名称
    private $age;                                   //年龄
    private $weight;                                //体重
    public function setName ($name)                 //设置名称
    {
        $this->name = $name;
    }
    public function getName ()                      //获得名称
    {
        return $this->name;
    }
    public function setAge ($age)                   //设置年龄
    {
        $this->age = $age;
    }
    public function getAge ()                       //获得年龄
    {
        return $this->age;
    }
    public function setWeight ($weight)             //设置体重
    {
        $this->weight = $weight;
    }
    public function getWeight ()                    //获得体重
    {
        return $this->weight;
    }
    public function __isset ($name)                 //定义__isset()方法
    {
        echo '在类中未定义属性' . $name;
    }
}
```

（2）使用 new 关键字对 Person 类进行实例化，然后分别调用该类中相应的 setXxx()方法为类中的属性赋初值，并使用函数 isset()判断类中未定义的 sex 属性是否被设置，实现该过程的代码如下所示：

```
require 'Person.php';                               //包含 Person.php 文件
$person = new Person();                             //实例 Person 类
$person->setName('小明');                           //设置名称
$person->setAge('12');                              //设置年龄
$person->setWeight('45 公斤');                      //设置体重
isset($person->sex);                                //判断是否在类中定义了 sex 属性
echo '<br/>';                                       //换行
echo $person->getName().'的年龄是：'.$person->getAge().'岁，体重是：'.$person->getWeight() ;    //打印信息
```

秘笈心法

心法领悟 341：使用 isset()函数判断变量是否被设置值。

PHP 编码时，有时需要判断某一变量是否被设置了值，然后再对该变量进行操作，PHP 中可以使用函数 isset()判断某个变量是否被设置了值，该函数的使用格式如下所示：

```
bool isset ( mixed var [, mixed var [, ...]] )
```

如果该函数的参数列表所指定的变量都已经设置了值则该函数返回 TRUE，反之返回 FALSE。

实例 342 使用__unset()方法提示未定义属性信息

光盘位置：光盘\MR\07\342

中级

趣味指数：★★★

实例说明

本实例主要讲解魔术方法__unset()的使用方法。运行本实例，如图 7.36 所示，在页面中打印出使用 unset()函数撤销类中未定义的价格属性的结果。

在Car类中未定义price属性

图 7.36 打印 Car 类中未定义的属性

关键技术

魔术方法__unset()是当使用 unset()函数撤销类中未定义的属性时所调用的，该方法的使用格式如下所示：

```
function __unset ($name)
{

}
```

魔术方法__unset()必须包含一个用于代表类中未定义属性的名称参数。

设计过程

（1）建立汽车类 Car，在类中定义汽车的品牌和颜色属性，然后使用构造方法为这两个属性赋初值，最后定义魔术方法__unset()打印撤销类中未定义属性时的提示信息，代码如下所示：

```
class Car
{
    private $brand;                                         //品牌
    private $color;                                         //颜色
    public function __construct ($brand, $color)            //构造方法
    {
        $this->brand = $brand;
        $this->color = $color;
    }
    public function __unset ($name)                         //定义__unset()方法
    {
        echo '在' . __CLASS__ . '类中未定义' . $name . '属性';
    }
}
```

（2）使用 require 语句包含 Car.php 文件，然后使用 new 关键字对汽车类进行实例化并生成汽车对象$car，最后使用 unset()方法撤销类中未定义的价格属性，代码如下所示：

```
require 'Car.php';                                          //包含 Car.php 文件
$car = new Car('奥迪', '黑色');                              //使用 new 关键字实例 Car 类
unset($car->price);                                         //使用函数 unset()释放 Car 类中未定义的属性 price
```

秘笈心法

心法领悟 342：PHP 中的魔术常量。

本实例中，使用__CLASS__获得当前类的名称，类似这样的常量在 PHP 中被称为魔术常量，其中 PHP 中常用的魔术常量及其说明如表 7.1 所示。

表 7.1　PHP 中常用的魔术常量

魔术常量名称	说　明
__LINE__	返回运行中 PHP 脚本的当前行的行号
__FILE__	返回当前 PHP 脚本的完整路径和文件名
__FUNCTION__	返回当前函数的名称
__CLASS__	返回当前类的名称
__METHOD__	返回当前类中成员方法的名称

实例 343　使用__autoload()方法自动导入类文件　中级

光盘位置：光盘\MR\07\343　趣味指数：★★★

实例说明

本实例主要讲解如何使用魔术方法__autoload()自动加载类文件。运行本实例，如图 7.37 所示，在页面中分别打印出苹果和桔子的颜色信息。

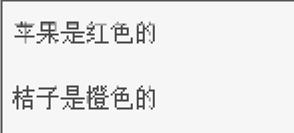


图 7.37　打印苹果和桔子的颜色属性

在实现本实例时，将 Apple 类和 Orange 类分别存放在与这两个类同名的类文件中，然后在 index.php 文件中不使用 require 等包含语句包含这两个类文件，而使用__autoload()方法根据类名自动导入相应的文件。

关键技术

魔术方法__autoload()包含一个参数，该参数是当前页面中所定义类的名称，在该魔术方法内部，可以使用 require 等包含语句和该方法的参数所代表的类名来自动加载类，__autoload()的使用格式如下所示：

```
function __autoload($name){

}
```

设计过程

（1）定义苹果类 Apple 和桔子类 Orange，并分别在这两个类中定义用于获得颜色的 getColor()方法，代码如下所示：

```
class Apple                                         //苹果类
{
    public function getColor ()                     //获得苹果颜色方法
    {
        return '苹果是红色的';
    }
}
class Orange                                        //桔子类
{
    public function getColor ()                     //获得桔子颜色的方法
    {
        return '桔子是橙色的';
    }
}
```

（2）定义__autoload()方法的实现。在该方法内部首先构建自动包含类文件的路径，然后使用函数 file_exists()判断该文件是否存在，如果存在，则使用 require_once 语句包含该文件，反之返回 FALSE 值，实现该过程的代码如下所示：

```
function __autoload ($className)                    //定义__autoload()方法
{
    $file = dirname(__FILE__) . '/class/' . $className . '.php';    //包含文件的路径
```

```
    if (! file_exists($file)) {                     //判断文件是否存在
        return false;                               //不存在返回 false
    } else {
        require_once $file;                         //存在则包含该文件
    }
}
```

（3）使用 require 语句包含 autoload.php 文件，然后使用 new 关键字分别对苹果类和桔子类进行实例化，最后通过这两个类实例的对象分别调用各自的 getColor()方法打印苹果和桔子的颜色，代码如下所示：

```
require 'autoload.php';                             //包含 autoload.php 文件
$apple = new Apple();                               //对苹果类进行实例化
$orange = new Orange();                             //对桔子类进行实例化
echo $apple->getColor() . '<br/>';                  //打印苹果颜色
echo $orange->getColor();                           //打印桔子颜色
```

秘笈心法

心法领悟 343：使用函数 file_exists()判断指定的文件是否存在。

本实例中，在魔术方法__autoload()内部使用函数 file_exists()判断指定的类文件是否存在，该函数的语法格式如下所示：

```
bool file_exists ( string filename )
```

如果 filename 指定的文件或目录存在则返回 TRUE，否则返回 FALSE。

7.8 设 计 模 式

面向对象的程序开发方式，不仅表现在程序的语法规则和语言特性上，更主要的是表现一种设计思想。当设计一个软件时，很多设计概念被反复使用，最终形成了一种规则，这些规则通常被称为设计模式。本节将通过具体实例讲解 PHP 常用设计模式的应用方式。

实例 344	使用单例模式制作数据库管理类 光盘位置：光盘\MR\07\344	中级 趣味指数：★★★

实例说明

本实例主要讲解如何使用单例模式制作一个基于 MySQL 数据库的数据库管理类。运行本实例，如图 7.38 所示，首先在用户登录文本框中输入用户名和密码，然后单击表单中的“登录”按钮，如果用户名和密码经后台代码验证正确则当前页面将跳转到登录成功信息提示页面，如果错误则在页面打印错误提示。在上述过程中，对数据库管理模块的开发采用 PDO 技术，设计上采用单例模式进行编码。



图 7.38　用户登录表单

关键技术

单例模式指的是在程序应用范围内只对指定的类创建一个实例，也就是说单例模式只包含一个对象，就是单例本身，PHP 使用单例模式设计的类通常应该满足如下要求：

☑ 单例模式的类通常拥有一个私有的构造方法和一个私有的克隆方法，这样可以确保用户无法通过创建对象或者克隆的方式对其进行实例化。

☑ 单例模式的类包含一个静态的用于保存当前类实例的成员变量，以及一个静态方法，该静态方法负责对其本身进行实例化，然后将实例化后的对象保存到类中所声明的静态变量中，从而确保一个对象被创建。

PHP 单例模式的基本形式如下所示：

```
class Singleton
{
    private static $instance;                                   //保存当前类实例后的对象
    public static function getInstance ()                       //静态方法，返回类实例
    {
        if (null == self::$instance) {                          //如果未定义
            self::$instance = new Db();
        }
        return self::$instance;                                 //返回当前对象
    }
    private function __construct ()                             //私有构造方法
    {}
    private function __clone ()                                 //私有克隆方法
    {}
     //类的其他方法
}
//获得类的对象
$singleton = Singleton::getInstance();
```

由上述代码可知，在类外部通过类名直接调用类中的用于返回当前类实例的静态方法即可获得当前类实例。

设计过程

（1）建立数据库连接类 Db，在类中定义用于保存当前类实例的静态成员变量$instance，并定义静态方法 getInstace()，在该方法内部，将当前类实例的对象赋给静态成员变量$instance 并返回，然后定义私有的构造方法和克隆方法，这样就构建了单例模式的基本框架，接下来创建 getConnId()方法，该方法用于返回使用 PDO 技术连接数据库后的句柄，最后创建数据库查询方法 query()实现对数据库的查询操作。实现上述过程的代码如下所示：

```
class Db                                                        //数据库类
{
    private static $instance;                                   //保存当前类实例后的对象
    public static function getInstance ()                       //静态方法，返回类实例
    {
        if (null == self::$instance) {                          //如果未定义
            self::$instance = new Db();
        }
        return self::$instance;
    }
    private function __construct ()                             //私有构造方法
    {}
    private function __clone ()                                 //私有克隆方法
    {}
    private function getConnId ()                               //获得数据库连接 ID 的方法
    {
        $config = parse_ini_file(dirname(__FILE__) . '/config.ini');               //解析配置文件
        $dsn = 'mysql:host=' . $config['host'] . ';dbname=' . $config['dbname'];   //配置数据源名称
        $pdo = new PDO($dsn, $config['username'], $config['password']);            //实例 PDO 对象
        $pdo->query('set names ' . $config['charset']);                            //设置字符集
        return $pdo;                                                               //返回 PDO 对象
```

```
	}
	public function query ($sql)                              //查询方法
	{
		$pdo = $this->getConnId();                            //获得 PDO 对象
		$sqlType = trim(substr($sql, 0, 6));                  //获得查询语句类型
		if ($sqlType == 'update' || $sqlType == 'delete' || $sqlType == 'insert') {
			return $pdo->query($sql);                         //如果是 update、delete 或 insert 语句则返回 SQL 语句执行结果
		} elseif ($sqlType == 'select') {                     //如果是 select 语句
			$tmpArray = array();                              //定义二维数组
			$results = $pdo->query($sql);                     //执行 select 语句
			foreach ($results as $result) {                   //将结果保存到二维数组中
				array_push($tmpArray, $result);
			}
			return $tmpArray;                                 //返回结果
		} else {
			return null;                                      //不是 update、delete、insert 或 select 语句，则返回 null
		}
	}
}
```

（2）建立用户登录表单，其实现过程请详见本书附带光盘。当用户在表单中输入用户名和密码，单击“提交”按钮后，将通过如下代码验证用户的登录信息是否正确：

```
if(isset($_POST['username']) && trim($_POST['username'])!='')
{
	require_once 'Db.php';
	$username = trim($_POST['username']);
	$password = trim(md5($_POST['password']));
	$db = Db::getInstance();
	$arrayUser = $db->query("select id from tb_user where username='".$username."' and password ='".$password."'");
	if(count($arrayUser)>0){
		$_SESSION['loginUsername'] =   $username;
		echo '<script>window.location.href="success.php";</script>';
	}else {
		echo '<div style="width:300px; height:30px; line-height:30px; border:1px solid #E59B04; background-color:#FCF2E0; color:#FF0000;">用户名或密码输入有误</div>';
	}
}
```

上述代码首先判断用户是否提交了表单，如果是，则通过 Db 类的 getInstance()方法获得类的实例，并通过该对象调用类中的 query()方法执行查询，根据查询结果的记录数来判断用户输入的用户名和密码是否正确，如果记录数大于 1，则说明用户登录信息有效。

秘笈心法

心法领悟 344：使用函数 parse_ini_file()解析.ini 文件。

在项目开发中，经常需要将项目的一些配置参数保存到.ini 文件中，例如本实例将数据库连接参数保存到专门的 config.ini 文件中，以便于对项目配置信息的修改和统一管理。函数 parse_ini_file()的语法格式如下所示：

```
array parse_ini_file ( string filename [, bool process_sections] )
```

函数 parse_ini_file()载入一个由 filename 指定的.ini 文件，并将其中的设置作为一个联合数组返回。如果 process_sections 参数设为 TRUE，将得到一个多维数组，包括配置文件中每一节的名称和设置，process_sections 的默认值是 FALSE。

实例 345	使用策略模式打印客户端浏览器类型 光盘位置：光盘\MR\07\345	中级 趣味指数：★★★

实例说明

本实例主要讲解在 PHP 面向对象编程中策略模式的应用方法。运行本实例，分别如图 7.39 和图 7.40 所示，其中图 7.39 所示为在 IE 下运行的结果，图 7.40 所示为在火狐浏览器下运行的结果。从运行结果可知，本实例

会根据不同浏览器的类型在页面中打印相应浏览器的名称。



图 7.39　打印浏览器的类型为 IE



图 7.40　打印浏览器的类型是 Firefox

关键技术

策略模式指的是程序中涉及决策控制的一种模式。策略模式通常定义一个抽象基类，然后根据不同情况创建不同的子类，其中一个简单的策略模式框架如下所示：

```
abstract class Strategy                              //定义抽象的基类
{
    abstract public function fun ();                 //定义抽象方法 fun()
}
class Child1 extends Strategy                        //定义子类 Child1，使其继承基类 Strategy
{
    public function fun ()                           //实现 fun()方法的功能
    {
        //Child1 中 fun()方法所实现的功能
    }
}
class Child2 extends Strategy                        //定义子类 Child2，使其继承基类 Strategy
{
    public function fun ()                           //实现 fun()方法的功能
    {
        //Child2 中 fun()方法所实现的功能
    }
}
if(条件 1==true){                                    //如果条件 1 为真
    $obj = new Child1();                             //生成子类 Child1 的实例对象
}else {
    $obj = new Child2();                             //生成子类 Child2 的实例对象
}
$obj->fun();
```

上述示例代码中首先定义抽象基类 Strategy，并在类中定义抽象方法 fun()，然后分别定义类 Child1 和 Child2，使这两个类分别继承基类 Strategy，在这两个子类的内部实现 fun()的功能，完成以上类的定义后，就可以根据不同的条件对不同的子类进行实例化。

设计过程

（1）定义抽象的 Browser 类，并在该类中声明抽象方法 getType()，然后定义子类 Ie 和 Fox，使这两个类分别继承基类 Browser，然后在类体内完成对 getType()类的定义，代码如下所示：

```
abstract class Browser                               //定义抽象类 Browser
{
    abstract public function getType ();
}
class Ie extends Browser                             //定义 Ie 类
{
    public function getType ()                       //获得类型方法
    {
        return 'IE';
    }
}
class Fox extends Browser                            //定义火狐类
{
    public function getType ()                       //获得类型方法
```

```
        {
            return 'Firefox';
        }
}
```

（2）使用 header()函数设置页面编码，并使用 require 语句包含 Browser.php 文件，然后定义 getBrowserType()方法获得浏览器类型，代码如下所示：

```
header('content-type:text/html; charset=utf-8');                //定义页面编码
require 'Browser.php';                                          //包含 Browser.php 文件
function getBrowserType ()                                      //定义获得浏览器类型的方法
{
    if (strstr($_SERVER['HTTP_USER_AGENT'], "IE")) {            //判断是否为 IE 浏览器
        $b = new Ie();                                          //实例 Ie 类
    } else {
        $b = new Fox();                                         //实例 Fox 类
    }
    return $b->getType();                                       //返回浏览器类型
}
echo '当前浏览器是' . getBrowserType();                          //执行 getBrowserType()方法
```

秘笈心法

心法领悟 345：使用函数 header()设置页面编码。

通过 HTML 语言的<meta>标记可以设置页面编码，那么如果在页面中不包含任何 HTML 标记的 PHP 页面如何设置页面编码呢？PHP 提供了用于发送 HTTP 头信息的 header()函数，可以通过如下方式设置页面编码：

```
header('content-type:text/html; charset=utf-8')
```

上述代码可以设置文件的内容类型为 text/html，页面编码为 utf-8 编码。

实例 346 使用工厂模式设置用户访问权限

光盘位置：光盘\MR\07\346

中级

趣味指数：★★★

实例说明

本实例主要讲解如何使用工厂模式对用户权限进行管理。运行本实例，如图 7.41 所示，在页面中将打印出不同用户的类型及各项操作的访问权限。制作本实例时，首先创建抽象的用户权限类，然后分别定义浏览、添加、更改和删除等子类，使之继承基类用户权限类，然后根据不同用户的权限，重写用户权限类中的方法，最后创建用户权限工厂来根据不同用户类别返回相应的对象。



用户名	用户类别	浏览	添加	更改	删除
大麦	访客	是	否	否	否
SOFT	会员	是	是	否	否
春晓	访客	是	否	否	否
金星	访客	是	否	否	否
360	会员	是	是	否	否
MR	管理员	是	是	是	是

图 7.41　电子相册用户权限列表

关键技术

工厂模式是指创建一个类似于工厂的类，通过对类中成员方法的调用返回不同类型的对象。使用工厂模式时，首先创建一个基类，然后根据对象类型的不同来创建不同的扩展类，最后定义工厂类来根据不同的条件返回相应的对象。工厂模式的示例代码如下所示：

```
abstract class Father                                   //定义基类
{

}
class Child1 extends Father                             //定义子类 1
{

}
class Child2 extends Father                             //定义子类 2
{

}
class Factory                                           //定义工厂类
{
    public static function create ($condition)          //创建对象
    {
        if ($condition == '条件 1') {
            return new Child1();
        } elseif ($condition == '条件 2') {
            return new Child2();
        }
    }
}
```

上述示例代码中，首先定义抽象基类 Father，然后定义子类 Child1 和 Child2，使之分别继承基类 Father，在实际应用中还应该重写或定义子类 Child1 和 Child2 中的方法，最后定义工厂类 Factory，并在该类中定义静态的 create()方法来根据不同的条件返回相应的对象。

提示：抽象基类也可以使用接口来代替。

设计过程

（1）定义用户权限基类 UserPermission，然后在该基类中定义 select()、add()、edit()和 delete()等方法，然后分别定义子类 GuestPermission、MemberPermission 和 AdminPermission，并使这些类分别继承基类 UserPermission，同时根据不同类的功能对基类中的方法进行重写，最后定义用户权限工厂 UserPermissionFactory 来根据不同用户的类别对相应的子类进行实例化并返回，实现该过程的代码如下所示：

```
abstract class UserPermission                           //定义抽象的用户权限类
{
    public function select ()                           //定义查询方法
    {
        return true;
    }
    public function add ()                              //定义添加方法
    {
        return false;
    }
    public function edit ()                             //定义编辑方法
    {
        return false;
    }
    public function delete ()                           //定义删除方法
    {
        return false;
    }
}
```

```
class GuestPermission extends UserPermission                    //定义浏览者权限类
{}
class MemberPermission extends UserPermission                   //定义会员权限类
{
    public function add ()                                      //重写添加方法
    {
        return true;
    }
}
class AdminPermission extends UserPermission                    //定义管理员权限类
{
    public function add ()                                      //重写添加方法
    {
        return true;
    }
    public function edit ()                                     //重写编辑方法
    {
        return true;
    }
    public function delete ()                                   //重写删除方法
    {
        return true;
    }
}
class UserPermissionFactory                                     //用户权限工厂
{
    public static function getUser ($userType)                  //获得用户权限方法
    {
        switch ($userType) {
            case 'GUEST':
                return new GuestPermission();                   //返回普通浏览者权限对象
                break;
            case 'MEMBER':
                return new MemberPermission();                  //返回会员权限对象
                break;
            case 'ADMIN':
                return new AdminPermission();                   //返回管理员权限对象
                break;
        }
    }
}
```

（2）定义二维数组$users，使用该数组保存用户名称及权限类别，代码如下所示：

```
$users = array(
    0=>array('name'=>'大麦', 'type' => 'GUEST'),
    1=>array('name'=>'SOFT', 'type' => 'MEMBER'),
    2=>array('name'=>'春泉', 'type' => 'GUEST'),
    3=>array('name'=>'金星', 'type' => 'GUEST'),
    4=>array('name'=>'360', 'type' => 'MEMBER'),
    5=>array('name'=>'MR', 'type' => 'ADMIN')
);
```

（3）使用 require 语句包含 UserPermissionFactory.php 及 db.php 文件，然后使用 foreach 循环语句遍历用户数组，并根据用户类别通过用户权限工厂生成相应的用户权限对象，最后打印用户权限信息，实现该过程的代码如下所示：

```
<?php
require 'UserPermissionFactory.php';                            //包含用户权限工厂所在文件
require 'db.php';                                               //包含用户信息数组所在文件
foreach ($users as $user){                                      //遍历用户权限数组
    $userPermission = UserPermissionFactory::getUser($user['type']);   //获得用户权限对象
?>
<!--  以下代码用于打印用户权限信息  -->
<tr>
    <td style="width:80px; height:22px; border:1px solid #E0056F;"><?=$user['name']?></td>
    <td style="width:80px; height:22px; border:1px solid #E0056F;"><?=str_replace('ADMIN', '管理员', str_replace('MEMBER', '会员',
str_replace('GUEST', '访客', $user['type'])))?></td>
    <td style="width:80px; height:22px; border:1px solid #E0056F;"><?=$userPermission->select()?'是':'否'?></td>
```

```
    <td style="width:80px; height:22px; border:1px solid #E0056F;"><?=$userPermission->add()?'是':'否'?></td>
    <td style="width:80px; height:22px; border:1px solid #E0056F;"><?=$userPermission->edit()?'是':'否'?></td>
    <td style="width:80px; height:22px; border:1px solid #E0056F;"><?=$userPermission->delete()?'是':'否'?></td>
</tr>
<?php
}
?>
```

秘笈心法

心法领悟 346：使用工厂模式设计数据库管理类。

根据本实例所介绍的工厂模式可知，如果采用该模式设计数据库管理类，可以使用相同的代码实现对不同数据库的管理。下面给出采用工厂模式设计的数据库管理类的模板代码：

```
abstract class Db                                   //定义抽象的数据库管理类
{
    abstract public function insert ();
    abstract public function delete ();
    abstract public function update ();
    abstract public function select ();
}
class MySQL extends Db                              //定义 MySQL 数据库管理类
{
    public function insert ()
    {}
    public function delete ()
    {}
    public function update ()
    {}
    public function select ()
    {}
}
class Oracle extends Db                             //定义 Oracle 数据库管理类
{
    public function insert ()
    {}
    public function delete ()
    {}
    public function update ()
    {}
    public function select ()
    {}
}
class SQLServer extends Db                          //定义 SQL Server 数据库管理类
{
    public function insert ()
    {}
    public function delete ()
    {}
    public function update ()
    {}
    public function select ()
    {}
}
class DbFactory                                     //定义数据库管理类工厂
{
    public static function getDb ($TYPE)            //根据不同的标识返回不同的数据库实例
    {
        switch ($TYPE) {
            case 'MYSQL':
                return new MySQL();
                break;
            case 'ORACLE':
                return new Oracle();
                break;
```

```
                case 'SQLSERVER':
                    return new SQLServer();
                    break;
            }
        }
    }
```

在上述模板代码中，首先定义抽象的数据库管理类 Db，然后分别定义子类 MySQL、Oracle 和 SQL Server，并定义数据库管理类工厂 DbFactory，在该工厂中定义静态的 getDb()方法，用来根据不同的标识返回数据库管理类对象，这样通过该对象就可以实现采用相同的方法对不同数据库进行管理。

第 3 篇

数据库篇

第 8 章

MySQL 数据库与 PHP

- MySQL 服务器的启动和关闭
- MySQL 数据库
- MySQL 数据表
- MySQL 数据
- PHP 操作 MySQL 数据库
- SQL 查询语句
- 多表查询
- 数据的导入导出
- MySQL 数据库的备份和恢复
- phpMyAdmin 图形化管理工具

8.1　MySQL 服务器的启动和关闭

PHP 在开发 Web 站点或一些管理系统时，需要对大量数据进行存储。XML 文件和文本文件虽然可以作为数据的载体，但不易进行管理和对大量数据的存储，所以在项目开发时，数据库就显得非常重要。PHP 可以连接的数据库种类较多，其中 MySQL 数据库与其兼容较好，在 PHP 开发中被广泛应用。

实例 347　启动 MySQL 服务器

光盘位置：光盘\MR\08\347

高级

趣味指数：★★★★

实例说明

如果想学习 PHP 语言，那么就必须要学 MySQL。PHP 与 MySQL 是最佳的组合，虽然 PHP 如今支持更多的数据库，例如 Access、SQL Server、Oracle、DB2 等，但是无论在 LAMP 组合（LINUX + Apache + MySQL + PHP），还是 AMP 组合（Apache + MySQL + PHP）中，MySQL 的地位丝毫没有动摇。

在 Windows 操作系统中，启动 MySQL 服务器可以使用命令模式进行。

关键技术

本实例的关键是在命令提示符下输入启动 MySQL 服务器的命令，完成 MySQL 服务器的启动，其命令如下：

```
net start mysql
```

设计过程

在 Windows 操作系统下启动 MySQL 服务器的步骤如下：

（1）选择“开始”/“运行”命令，在弹出的“运行”对话框的“打开”文本框中输入 cmd 命令，按 Enter 键进入命令提示符窗口，如图 8.1 所示。

（2）在命令提示符下输入启动 MySQL 服务器的命令，如图 8.2 所示。



图 8.1　启动 MySQL 服务器第 1 步



图 8.2　成功启动 MySQL 服务器

秘笈心法

心法领悟 347：手动启动 MySQL 服务。

启动 MySQL 服务器还有另外一种方法，即通过选择“开始”/“控制面板”/“管理工具”/“服务”命令，打开 Windows 服务管理器，在服务器的列表中找到“mysql”服务，单击鼠标右键，从弹出的快捷菜单中选择“启动”命令即可。

实例 348 连接 MySQL 服务器 高级

光盘位置：光盘\MR\08\348 趣味指数：★★★★

实例说明

启动 MySQL 服务只是应用 MySQL 的第一步，如果想要操作 MySQL 数据库，就要学会连接 MySQL 服务器。本实例详细讲解如何在命令提示符下连接 MySQL 服务器。

关键技术

选择“开始”/“运行”命令，输入 cmd 命令，进入命令提示符提示框，在其中输入如下命令：

```
mysql –uName –p Password
```

“mysql”与“-u”、“Name”与“-p”之间有一个空格。Name 表示进入 MySQL 服务器的用户名；Password 表示进入 MySQL 服务器的密码。

设计过程

连接 MySQL 服务器的步骤如下：

（1）进入命令提示符窗口。

（2）如果 MySQL 默认安装在 C:\盘根目录下，在未设置环境变量的情况下，须在命令提示符下进入 mysql\bin 文件夹，输入命令。假设用户名为 root，密码为 111，输入的命令如下：

```
mysql –uroot –p111
```

按 Enter 键，输出如图 8.3 所示的内容，表示连接 MySQL 成功。

（3）如果在本地用户机上运行程序，可以直接输入命令 mysql -uroot -p111 来连接 MySQL，但是为了保证安全，避免他人得到连接 MySQL 数据库的密码进行非法操作，最好使用如下命令：

```
mysql –uroot -p
```

并根据提示输入密码完成连接，如图 8.4 所示。

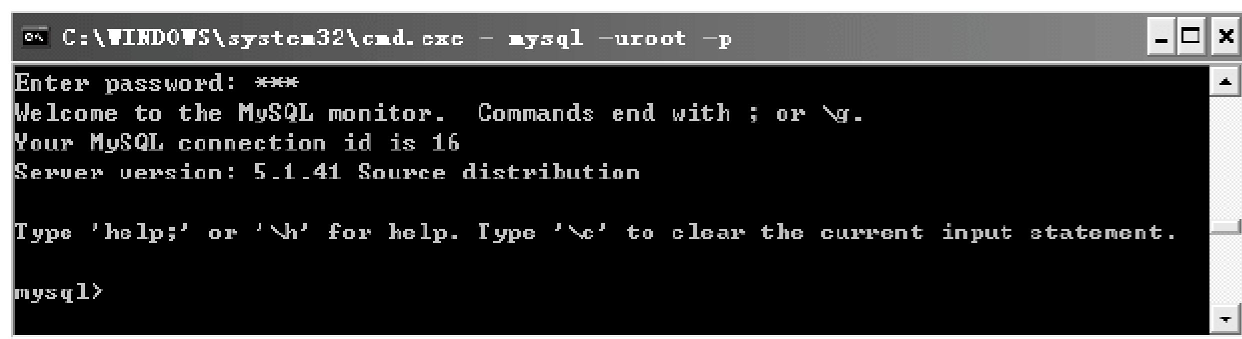


图 8.3 连接 MySQL 成功

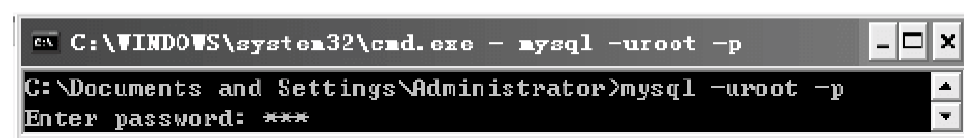


图 8.4 连接 MySQL 成功

秘笈心法

心法领悟 348：设置 MySQL 环境变量。

（1）在连接 MySQL 数据库时，可以通过设置环境变量 PATH 连接 MySQL，这样一来用户无需手动进入 MySQL 的 bin 文件夹，可以直接输入命令连接 MySQL。设置 MySQL 环境变量的步骤如下：

① 右键单击“我的电脑”图标，选择“属性”/“高级”/“环境变量”命令。

② 在 Administrator 的用户变量下选择“PATH”变量，单击“编辑”，在弹出的文本框中添加 MySQL 安装文件中 bin 文件夹的位置，例如“;C:\mysql\bin”，单击“确定”按钮，环境变量配置完毕。

③ 在命令提示符下输入如下命令：mysql -uroot -p111。

④ 按 Enter 键，输出如图 8.3 所示的内容，连接 MySQL 成功。

实例 349　关闭 MySQL 服务器　高级

光盘位置：光盘\MR\08\349　趣味指数：★★★★

实例说明

为了有效节省系统资源，在进行 MySQL 数据库操作后，需要及时将 MySQL 服务关闭。关闭 MySQL 服务的方法与启动 MySQL 服务相似，可以通过两种方法实现。通过本实例，相信用户会将关闭 MySQL 服务器的两种方法熟记于心。

关键技术

在命令提示符下，输入命令：

（1）断开 MySQL 数据库连接。

```
exit
```

（2）关闭 MySQL 服务。

```
net stop mysql
```

设计过程

（1）在命令提示符下，输入命令连接 MySQL，通过 MySQL 数据库函数“user”显示数据表中的内容，如图 8.5 所示。

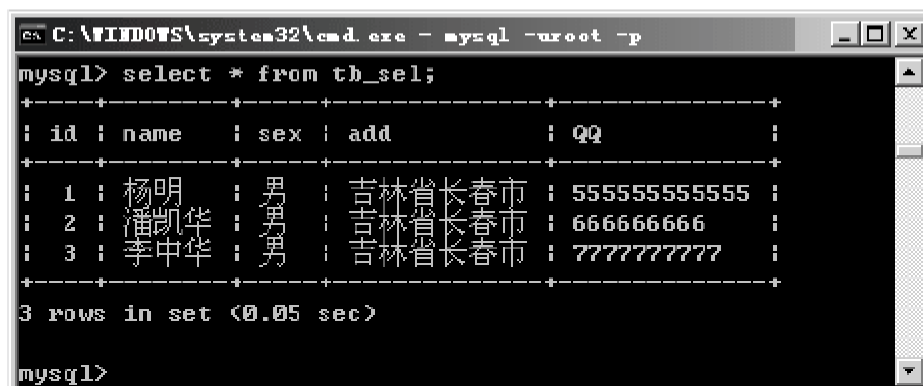


图 8.5　查询表信息

（2）通过命令 exit 断开与 MySQL 的连接，如图 8.6 所示。

（3）输入命令 net stop mysql，按 Enter 键，关闭 MySQL 服务，如图 8.7 所示。

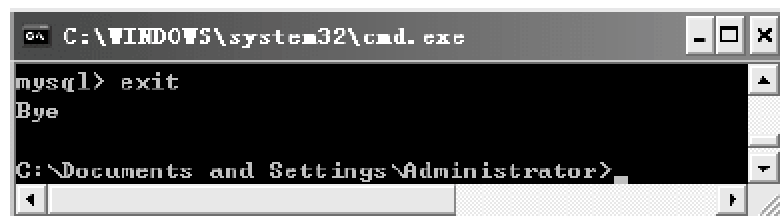


图 8.6　断开与 MySQL 连接



图 8.7　关闭 MySQL 服务

秘笈心法

心法领悟 349：手动停止 MySQL 服务。

关闭 MySQL 服务还有另外一种方法，即通过选择“开始”/“控制面板”/“管理工具”/“服务”命令，打开 Windows 服务管理器，在服务列表中找到“mysql”服务，单击鼠标右键，从弹出的快捷菜单中选择“停止”命令即可，如图 8.8 所示。

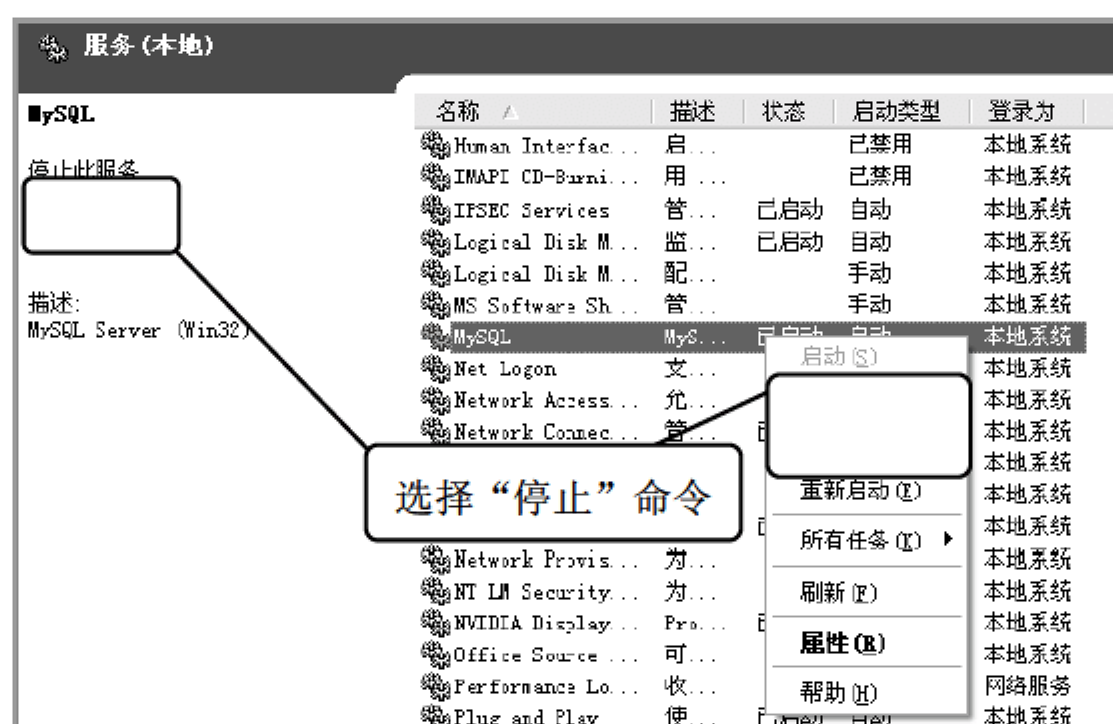


图 8.8　在 Windows 服务管理器中关闭 MySQL 服务

8.2　MySQL 数据库

MySQL 是一款安全、跨平台、高效，并与 PHP、Java 等主流编程语言紧密结合的数据库系统。该数据库系统由瑞典的 MySQL AB 公司开发、发布并支持，由 MySQL 的初始开发人员 David Axmark 和 Michael Monty Widenius 于 1995 年建立。MySQL 的象征符号是一只名为 Sakila 的海豚，代表着 MySQL 数据库的速度、能力、精确和优秀本质。

实例 350　创建 PHP 图书数据库

光盘位置：光盘\MR\08\350

高级

趣味指数：★★★★

实例说明

如果用户已经掌握如何启动、连接、断开和关闭 MySQL 服务，下面就走进 MySQL 数据库，学习如何创建数据库。 创建数据库的主要目的是将同类的或者是同一个项目中的数据表进行分类存储，方便开发人员查找和使用。本实例通过数据库函数“CREATE DATABASE”创建 PHP 图书数据库。

关键技术

（1）创建数据库函数。

```
CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db_name;
```

CREATE DATABASE 用于创建数据库，并进行命名。参数 db_name 表示需要创建的数据库名称。

（2）获取数据库列表。

```
SHOW {DATABASES | SCHEMAS} [LIKE 'pattern'];
```

SHOW DATABASES 可以列举数据库名称。

设计过程

（1）进入命令提示符窗口，启动 MySQL 服务，连接 MySQL 服务器。

（2）通过数据库函数在命令提示符下编写命令，代码如下：

```
CREATE DATABASE DB_MRBOOK;
```

按 Enter 键，运行结果如图 8.9 所示。

（3）查看 MySQL 数据库列表，查找创建的 PHP 图书数据库是否存在，运行结果如图 8.10 所示。

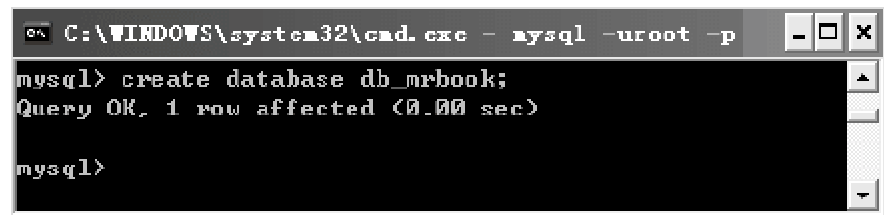


图 8.9　创建数据库 db_mrbook

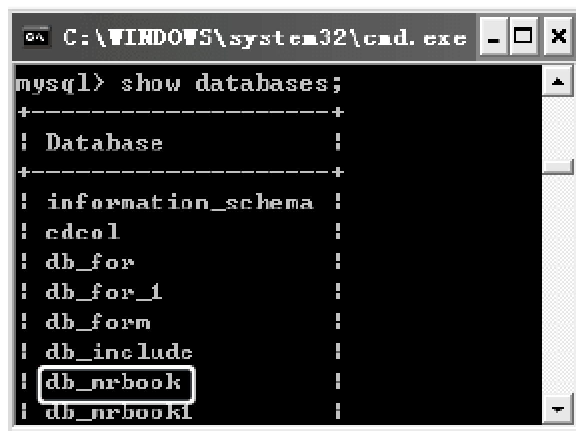


图 8.10　数据库 db_mrbook 存在于数据库列表中

（4）断开数据库连接，关闭 MySQL 服务。

秘笈心法

心法领悟 350：数据库命名规则。

在创建数据库时，数据库命名要遵循以下几项规则。

☑ 不能与其他数据库重名，否则将发生错误。

☑ 名称可以由任意英文字母、阿拉伯数字、下划线或者$组成，可以使用上述的任意字符开头，但不能使用单独的数字，否则会造成数据库名与数值相混淆。

☑ 名称最长可为 64 个字符，而别名最长可达 256 个字符。

☑ 不能使用 MySQL 关键字作为数据库、表名。

☑ 默认情况下，Windows 下是不区分数据库名、表名的大小写的，而在 Linux 下要区分数据库名、表名的大小写。为了便于数据库在平台间进行移植，建议采用小写来定义数据库名和表名。

实例 351	选择 PHP 图书数据库 光盘位置：光盘\MR\08\351	高级 趣味指数：★★★★

实例说明

创建数据库的目的是向数据库中存储同类型或者同一个项目下的所有数据表。若用户已经创建了数据库，需要通过命令选择指定的数据库，否则计算机将无法自动识别创建的数据表属于哪个数据库，还会出现错误提示，如图 8.11 所示。

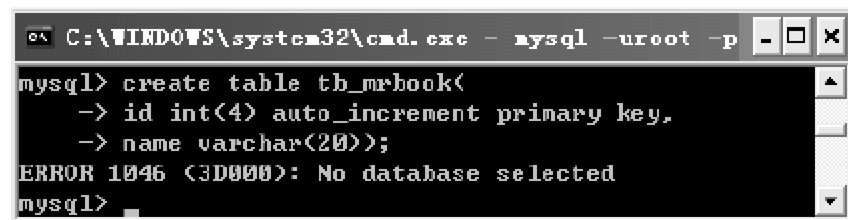


图 8.11　错误提示

本实例通过数据库函数“USE db_name”实现选择 PHP 图书数据库。

关键技术

使用 USE db_name 语句可以通过 MySQL 把 db_name 数据库作为默认（当前）数据库，并应用于后续语句。该数据库保持为默认数据库，直到语段的结尾，或者直到发布一个不同的 USE 语句，语法格式如下：

```
USE db_name;
```

参数 db_name 表示需要使用的数据库名称，该名称必须是合法的。

设计过程

（1）在命令提示符窗口下，通过命令启动 MySQL 服务，连接 MySQL 服务器。创建数据库并将此数据库命名为 db_mrbook。

（2）通过 show databases;语句，查看在 MySQL 服务器主机上列举的数据库名称是否包含用户指定的数据库。

（3）输入命令 use db_mrbook;，将数据库 use db_mrbook 作为当前默认数据库，按 Enter 键，命令窗口将输出如图 8.12 所示的内容。

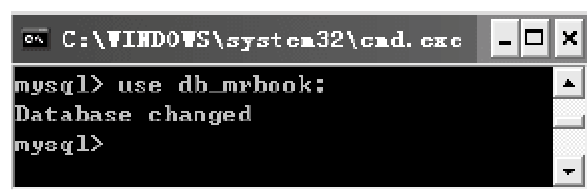


图 8.12　选择数据库

（4）关闭数据库连接，关闭 MySQL 服务。

秘笈心法

心法领悟 351：USE 语句的特点。

使用 USE 语句为一个特定的当前的数据库做标记，不会阻碍访问其他数据库中的表，例如：

```
use db1;
select * from db2.tb2;
```

在 cmd 命令提示符下，把数据库 db1 作为当前数据库标记，访问数据库 db2 下表 tb2 中所有的数据，这样做也是正确的。

实例 352　删除 PHP 图书数据库

光盘位置：光盘\MR\08\352

高级

趣味指数：★★★★

实例说明

若某个数据库失去了存在的价值，那么为了节省系统硬盘资源，提高 MySQL 的运行效率，建议将其从硬盘上删除。删除数据库可以通过打开 MySQL 安装目录，进入 data 文件夹，删除指定的数据库文件夹，还可以通过在命令提示符下输入删除命令。本实例讲解在命令提示符下输入命令，实现删除 PHP 图书数据库。

关键技术

DROP DATABASE 语句用于删除数据库中的所有表格和数据库。使用此语句需要获得数据库 DROP 权限，其语法格式如下：

```
DROP {DATABASE | SCHEMA} [IF EXISTS] db_name;
```

设计过程

（1）进入命令提示符窗口，启动 MySQL 服务，连接 MySQL 服务器。

（2）通过命令语句 show databases;查看在 MySQL 服务器主机上列举的数据库名称是否存在数据库 db_mrbook。

（3）使用命令语句 use db_mrbook;为数据库 db_mrbook 做上标记，使其当前默认数据库，如图 8.13 所示。

（4）在命令行输入 drop database db_name;语句，删除数据库的同时也删除数据库中的所有表，所以在命令行输入命令删除数据库时，最好清查数据库中的所有表名，以防止意外删除运行程序中的作用表而造成不必要的损失。查看数据库中所有表名可以通过命令语句 show tables;实现，如图 8.14 所示。

（5）使用命令行语句删除 PHP 图书数据库 db_mrbook，如图 8.15 所示。

（6）断开与 MySQL 服务器的连接，关闭 MySQL 服务。

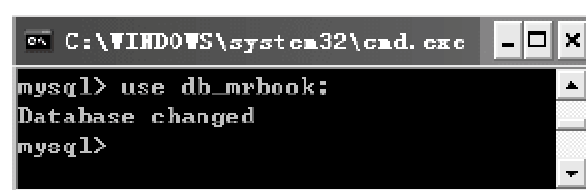


图 8.13　选择数据库

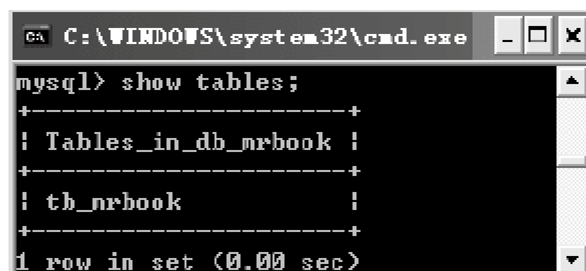


图 8.14　显示数据库中的数据表



图 8.15　删除数据库

秘笈心法

心法领悟 352：屏蔽删除不存在数据库提示。

在命令提示符窗口中，输入删除不存在的数据库命令，会发生 1008 错误，提示无法删除指定数据库，因为指定数据库不存在。如果用户想屏蔽此错误提示，可以使用数据库删除语句 drop database db_name;中的可选参数 if exists。例如输入命令语句 drop database if exists db_name;，则无论是否存在 db_name 数据库，命令行会一直提示操作成功，如图 8.16 所示。

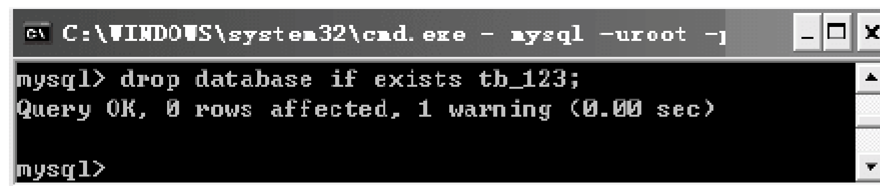


图 8.16　删除不存在数据库

8.3　MySQL 数据表

创建完数据库后，就可以在命令提示符下对数据库进行操作了，如创建数据表、更改数据表结构以及删除数据表等。

实例 353	在 PHP 图书数据库中创建图书信息表 光盘位置：光盘\MR\08\353	高级 趣味指数：★★★★

实例说明

数据库的主要作用是保存数据表，而数据表的作用是保存数据信息。本实例通过 CREATE TABLE 语句实现在 PHP 图书数据库中创建图书信息表，如图 8.17 所示。

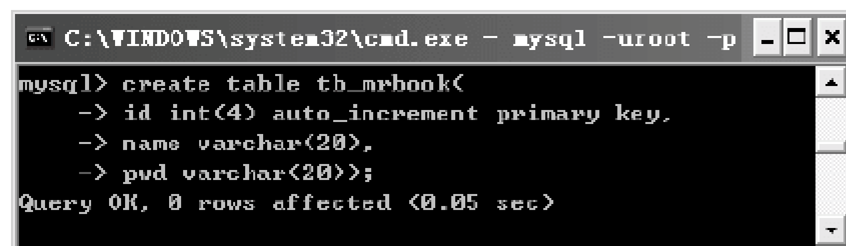


图 8.17　创建数据表

关键技术

创建数据表使用 CREATE TABLE 语句，其语法格式如下：

```
CREATE TABLE table_name
(create  definition,….)
[table_opotions]
[select_statement]
```

参数说明：

table_name：要创建的数据表名。

create_definition：表的列属性部分。MySQL 要求创建的表至少包含一列。

table_options：表的一些特性参数。

select_statement：SELECT 语句描述部分，可以快速地创建表。

下面介绍列属性部分，每一列定义的具体格式如下，代码如下：

```
col_name type[NOT NULL | NULL][DEFAULT default_value][AUTO_INCREMENT][PRIMARY KEY]
```

参数说明：

col_name：表示字段名。

type：表示字段类型。

NOT NULL 或者 NULL：指出该列是否允许是空值。所谓的空值是“不知道”或“无意义”的值，但是数据“0”和空格都不是空值，系统一般默认允许为空值，所以当不允许为空值时，必须使用 NOT NULL。

DEFAULT default_value：表示默认值。

AUTO_INCREMENT：表示是否是自动编号，每个表只能有一个 AUTO_INCREMENT 列，并且必须被索引。

PRIMARY KEY：表示是否为主键。它是一个唯一的 KEY，还有一个额外的约束，即所有键列必须被定义

为 NOT NULL。在 MySQL 中，该列被命名为 PRIMARY。一个表只能有一个 PRIMARY KEY。如表中没有 PRIMARY KEY，而某些应用程序需要时，MySQL 将返回第一个没有任何 NULL 列的 UNIQUE 键，作为 PRIMARY KEY。一个 PRIMARY KEY 可以是一个多列索引，但是不能在一个列规格说明中使用 PRIMARY KEY 键属性来创建一个多列索引，这样做将仅仅标记单个列作为主键，必须使用 PRIMARY KEY(index_col_name,…)句法。如果 PRIMARY KEY 或 UNIQUE 键只由一个列组成，并且列类型是整型，则可以用_rowid 引用。

以上是创建一个数据表的一些基础知识，它看起来十分复杂，在实际的应用中，使用最基本的格式创建数据表即可，具体格式如下：

```
create table table_name(列名 1 属性，列名 2 属性…)
```

设计过程

（1）启动 MySQL 服务，建立 MySQL 服务器连接。

（2）在命令提示符窗口下通过命令语句 create database db_name;创建名称为 db_mrbook 的数据库。

（3）使用 use db_name;语句设定数据库标记。

（4）通过 create table tb_name();语句创建数据表，代码如下：

```
mysql> use db_mrbook;
Database changed
mysql> create table tb_mrbook(
    -> id int(4) auto_increment prima
    -> name varchar(20) NOT NULL,
    -> pwd varchar(20) NOT NULL);
Query OK, 0 rows affected (0.05 sec)
```

（5）断开 MySQL 服务器连接，结束 MySQL 服务。

秘笈心法

心法领悟 353：SQL 语句格式。

在输入 SQL 语句时，可以一行全部输出，也可以每个字段都换行输出，这里建议用换行输出，因其看上去美观、易懂，且在语句出现错误时更容易查找。

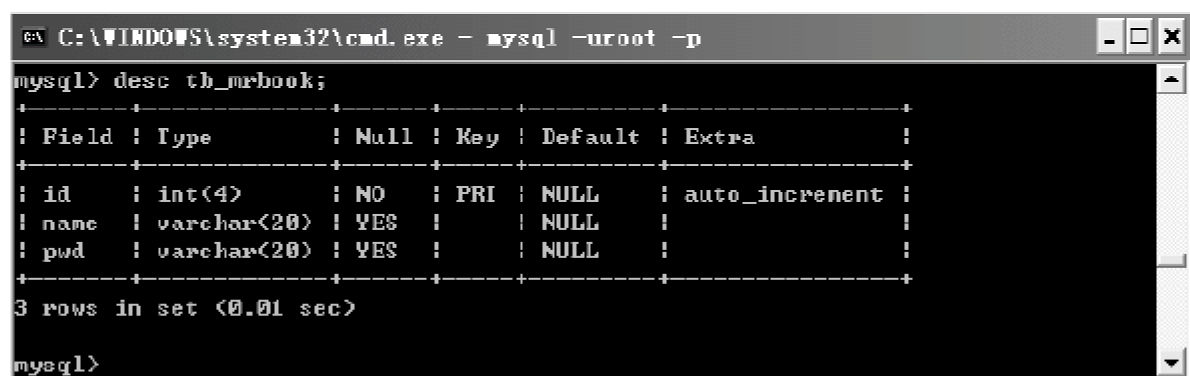
实例 354　查看图书信息表

光盘位置：光盘\MR\08\354

高级　趣味指数：★★★★

实例说明

查看信息表分两种形式：第一种是查看信息表结构，第二种是查看信息表数据。实例 353 通过 create table tb_name;创建的只是表结构，并没有插入数据信息，如果想查看信息表数据信息，需要使用 select 查询语句。本实例通过 show columns 命令查看信息表结构，运行结果如图 8.18 所示。



```
C:\WINDOWS\system32\cmd.exe - mysql -uroot -p
mysql> desc tb_mrbook;
+-------+-------------+------+-----+---------+----------------+
| Field | Type        | Null | Key | Default | Extra          |
+-------+-------------+------+-----+---------+----------------+
| id    | int(4)      | NO   | PRI | NULL    | auto_increment |
| name  | varchar(20) | YES  |     | NULL    |                |
| pwd   | varchar(20) | YES  |     | NULL    |                |
+-------+-------------+------+-----+---------+----------------+
3 rows in set (0.01 sec)

mysql>
```

图 8.18　查看信息表结构图

关键技术

使用 show columns 命令查看数据表结构，其语法如下：

```
show [full] columns  from 数据表名 [from 数据库名];
```

或写成

```
show  [full] columns  FROM 数据表名.数据库名;
```

设计过程

（1）开启 MySQL 服务，在命令提示符窗口输入命令，创建 MySQL 连接。通过 create database db_name;和 create table tb_name;语句创建数据库和数据表。

（2）通过命令语句 show columns from tb_name;查看表结构，其代码如下：

```
mysql> show columns from tb_mrbook;
```

（3）通过 exit 和 net stop mysql;语句断开数据库服务器，并关闭 MySQL 服务。

秘笈心法

心法领悟 354：describe 命令的特点。

查看数据表结构还可以通过 describe 命令。describe 命令的语法格式如下：

```
describe 数据表名;
```

其中，describe 可以简写成 desc。在查看表结构时，也可以只列出某一列的信息，其语法格式如下：

```
describe 数据表名 列名;
```

实例 355　修改图书信息表

光盘位置：光盘\MR\08\355

高级

趣味指数：★★★★

实例说明

程序员在建立数据表时，难免在表的结构上出现错误，例如将 Date 类型数据设置为 int 类型，这时就需要更改表的结构。本实例通过增加表 tb_mrbook 中的 last_date 字段，并设置数据类型为 Date 实现修改信息表，运行结果如图 8.19 所示。

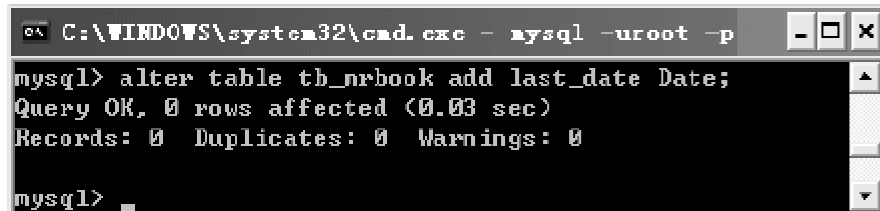


图 8.19　修改图书信息表

关键技术

修改表结构是指增加或者删除字段、修改字段名称或者字段类型、设置取消主键外键、设置取消索引以及修改表的注释等。修改表结构采用 alter table 命令，其语法如下：

```
alter [IGNORE] table 数据表名 alter_spec[,alter_spec]…;
```

注意，当指定 IGNORE 时，如果出现重复的行，则只执行一行，其他重复的行被删除。其中，alter_spec 子句定义要修改的内容，其语法如下：

```
alter_specification:
    ADD [COLUMN] create_definition [FIRST | AFTER column_name ]          --添加新字段
  | ADD INDEX [index_name] (index_col_name,...)                          --添加索引名称
  | ADD PRIMARY KEY (index_col_name,...)                                --添加主键名称
  | ADD UNIQUE [index_name] (index_col_name,...)                         --添加唯一索引
  | ALTER [COLUMN] col_name {SET DEFAULT literal | DROP DEFAULT}         --修改字段名称
  | CHANGE [COLUMN] old_col_name create_definition                      --修改字段类型
  | MODIFY [COLUMN] create_definition                                   --修改子句定义字段
  | DROP [COLUMN] col_name                                              --删除字段名称
  | DROP PRIMARY KEY                                                    --删除主键名称
  | DROP INDEX index_name                                               --删除索引名称
  | RENAME [AS] new_tbl_name                                            --更改表名
  | table_options
```

设计过程

（1）开启 MySQL 服务器，在命令提示符窗口下创建 MySQL 连接。通过 create database db_name;和 create table tb_name;语句创建数据库和数据表。

（2）使用命令语句 show columns from tb_name;查看表结构。

（3）在命令提示符下输入如下命令：

```
mysql> alter table tb_mrbook add last_date Date;
```

按 Enter 键，运行结果如图 8.19 所示。

（4）通过 exit 和 net stop mysql;语句断开数据库服务器并关闭 MySQL 服务。

秘笈心法

心法领悟 355：alter table 语句的特点。

alter table 语句允许指定多个动作，其动作间使用逗号分隔，每个动作表示对表的一个修改。

实例 356 重命名图书信息表

光盘位置：光盘\MR\08\356

高级

趣味指数：★★★★

实例说明

MySQL 数据表是不允许重命名的，所以在同一个数据库中绝对不会出现两个名称相同的表，但是一般情况下，程序员希望自己创建的数据表更能贴近自己的项目，所以对数据表重命名有时是不可避免的。本实例通过 rename table 命令实现对图书信息表重命名，运行结果如图 8.20 所示。

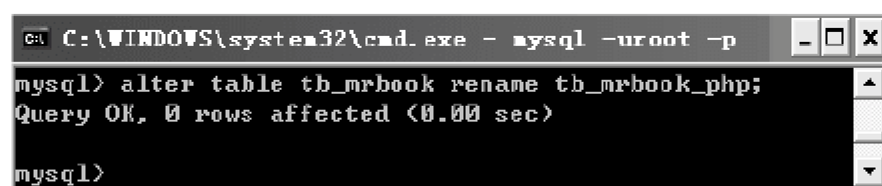


图 8.20 重命名图书信息表

关键技术

重命名数据表采用 rename table 命令，语法格式如下：

```
rename table 数据表名 1 To 数据表名 2;
```

或者

```
alter table 数据表名 1 rename 数据表名 2;
```

设计过程

（1）在命令提示符窗口下开启 MySQL 服务器服务，连接 MySQL 数据库，创建数据库并建立数据表。

（2）使用命令语句 show tables;显示当前数据库中的所有表。

（3）在命令提示符下输入如下命令：

```
mysql> alter table tb_mrbook rename tb_mrbook_php;
```

按 Enter 键，完成数据表的重命名操作，运行结果如图 8.20 所示。

（4）断开 MySQL 服务器连接，关闭 MySQL 服务。

秘笈心法

心法领悟 356：rename 语句特点。

使用“rename table 数据表名 1 To 数据表名 2”语句可以同时对多个数据表进行重命名，多个表之间以逗号“,”分隔。

实例 357　删除图书信息表

光盘位置：光盘\MR\08\357

高级

趣味指数：★★★★

实例说明

使用 drop database db_name;语句删除数据库的同时会连带删除数据库中的所有数据表，如果只想删除数据库中的某个表而保留其他所有表，显然此语句已经不再适用。删除数据库中某个数据表可以使用 drop table tb_name;语句，运行结果如图 8.21 所示。

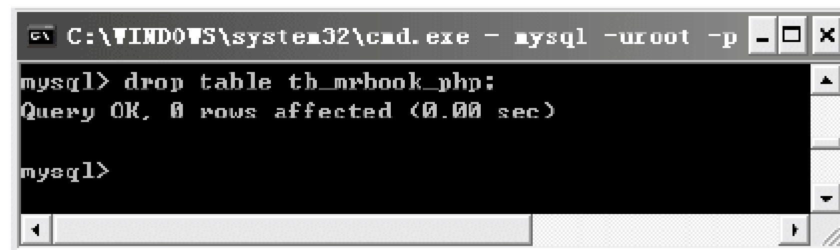


图 8.21　删除数据库中指定数据表

关键技术

删除数据表的操作很简单，同删除数据库的操作类似，使用 drop table 命令就可以实现，其语法格式如下：

```
drop table 数据表名;
```

设计过程

（1）在命令提示符窗口下开启 MySQL 服务，通过输入命令连接到 MySQL 服务器。

（2）使用命令语句 show tables;显示当前数据库中的所有表。

（3）在命令提示符下输入如下命令，删除指定数据表。

```
drop table tb_mrbook;
```

按 Enter 键，运行结果如图 8.21 所示。

（4）断开 MySQL 服务器连接，关闭 MySQL 服务。

秘笈心法

心法领悟 357：屏蔽删除不存在数据表的错误提示。

在删除数据表的过程中，如果删除一个不存在的表将会产生错误，这时在删除语句中加入 if exists 关键字，就不会出错，其语法格式如下：

```
drop table if exists 数据表名;
```

8.4　MySQL 数据

前面介绍了对数据库和数据表的操作，它们是包含与被包含的关系，即数据库包含数据表，数据表被包含于数据库。而数据表中包含数据，在 MySQL 数据库中真正被调用的是某个数据库中数据表里面的数据，因此如何更好地操作和使用这些数据才是使用 MySQL 数据库的根本。

实例 358　向图书信息表中添加数据

光盘位置：光盘\MR\08\358

高级

趣味指数：★★★★

实例说明

建立数据库是为了把不同类型或者不同要求的数据分类存储在不同的数据表中，即创建数据表是为了将相同类型或者相同要求的数据存储在同一表中。本实例讲解通过 SQL 语句向图书信息表中添加数据，运行结果如图 8.22 所示。



图 8.22　向图书信息表中添加数据

关键技术

建立一个空的数据表时，首先要想到的就是如何向数据表中添加数据。这项操作可以通过 insert 命令来实现。insert 语句的语法如下：

```
insert into 数据表名(column_name,column_name2, … ) values (value1, value2, … );
```

设计过程

（1）在命令提示符窗口下开启 MySQL 服务，通过命令行语句连接 MySQL 服务器，创建数据库并建立数据表。

（2）编辑 insert 语句，执行数据的添加操作，其代码如下：

```
insert into tb_mrbook_php(id,name,pwd,last_date)values ('','lizhonghua','5315',now());
```

（3）当提示符窗口出现如图 8.23 所示的内容时，表示数据添加成功。

图 8.23　数据添加成功

（4）通过 select 语句查询数据表中内容，显示结果如图 8.22 所示。

（5）断开 MySQL 服务器连接，关闭 MySQL 服务。

秘笈心法

心法领悟 358：insert 语句的特点。

在 MySQL 中，一次可以同时插入多行记录，各行记录的值在 values 关键字后以逗号“,”分隔，而标准的 SQL 语句一次只能插入一行。

实例 359　修改图书信息表中的数据

光盘位置：光盘\MR\08\359

高级　趣味指数：★★★★

实例说明

用户向数据表中插入数据信息时，若在手动输入过程中出现错误，就需要使用 SQL 语句对错误的数据进行

修改。本实例通过修改语句，修改图书信息表中操作数据表的时间，运行结果如图 8.24 所示。



图 8.24　修改图书信息表

关键技术

执行修改操作应用的是 update 命令，该语句的语法格式如下：

```
update 数据表名 set column_name = new_value1,column_name2 = new_value2, …where condition
```

其中，set 子句指出要修改的列及其给定的值；where 子句是可选的，如果给出该子句将指定记录中哪一行应该被更新，否则，所有的记录行都将被更新。

设计过程

（1）在命令提示符窗口下开启 MySQL 服务，通过命令行语句连接 MySQL 服务器，创建数据库并建立数据表。

（2）使用 select 查询语句查询图书信息表，如图 8.24 中❶原数据表所示。

（3）使用 update 更新语句更新数据表中数据，其代码如下：

```
update tb_mrbook_php set last_date='2010-07-11' where id=1;
```

（4）当提示符窗口出现如图 8.25 所示的内容时，表示数据更新成功。

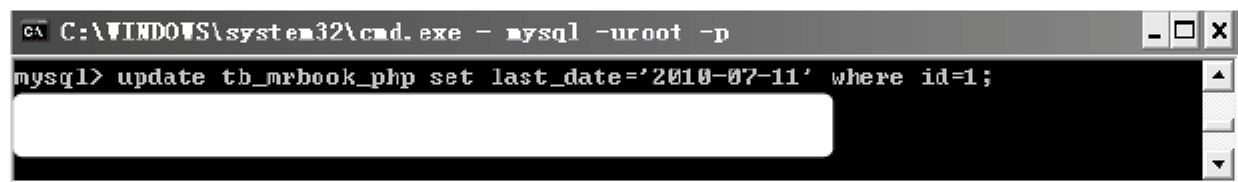


图 8.25　修改成功提示

（5）通过 select 语句查询数据表中的内容，显示结果如图 8.24 中❸修改后数据所示。

（6）断开 MySQL 服务器连接，关闭 MySQL 服务。

秘笈心法

心法领悟 359：使用 where 子句的注意事项。

修改数据信息时一定要保证 where 子句的正确性，一旦 where 子句出错，有可能会破坏所有的数据信息。

实例 360

删除图书信息表中所有数据

光盘位置：光盘\MR\08\360

高级

趣味指数：★★★★

实例说明

删除数据表中的所有数据在实际开发中是不多见的，删除数据表中的所有数据意味着清空数据表中的所有信息，如果是操作上的失误造成数据表中所有信息丢失，对于程序员来说是灾难性的，所以对数据表中信息的删除操作需要十分谨慎。本实例通过两种方法实现删除图书信息表中所有数据，运行结果如图 8.26 和图 8.27 所示。

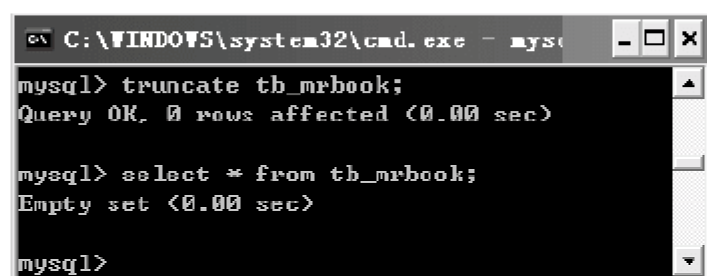


图 8.26　清空数据库

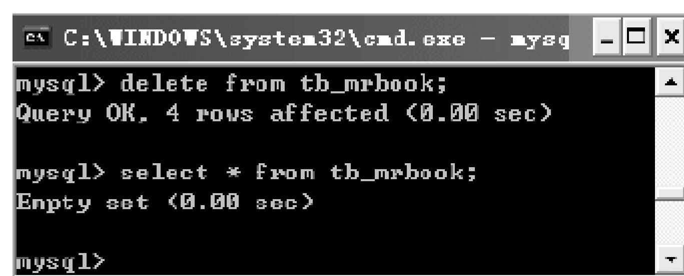


图 8.27　delete 语句删除信息

关键技术

（1）使用 delete 语句删除数据信息，其语法如下：

```
DELETE [LOW_PRIORITY] [QUICK] [IGNORE] FROM tbl_name
    [WHERE where_definition]
    [ORDER BY ...]
    [LIMIT row_count]
```

该语句在执行过程中，删除 tbl_name 表中的记录，如果没有指定 where 条件，将删除所有的记录；如果指定 WHERE 条件，将按照指定的条件进行删除。参数 ORDER BY 表示按条件排序，参数 LIMIT row_count 表示控制显示条数。

（2）使用 truncate 语句清空数据表信息，其语法如下：

```
TRUNCATE [TABLE] tb_name;
```

用于完全清空数据表，从逻辑上说，该语句与用于删除所有行的 delete 语句等同。

设计过程

（1）开启 MySQL 服务，通过命令提示符窗口创建 MySQL 连接，新建数据库和数据表，将数据表命名为 tb_mrbook，利用 INSERT 插入语句向数据表中插入数据，并通过 select 语句查询数据表。

（2）利用 truncate 语句，清空数据表中所有数据，其代码如下：

```
truncate tb_mrbook;
```

运行结果如图 8.26 所示。

（3）利用 delete 语句删除所有数据，其代码如下：

```
delete from tb_mrbook;
```

运行结果如图 8.27 所示。

秘笈心法

心法领悟 360：按 id 删除数据。

在实际的应用中，在执行删除操作时，执行删除的条件一般应该为数据的 id，而不是具体某个字段值，这样可以避免一些不必要的错误发生。

实例 361　删除图书信息表中指定数据

光盘位置：光盘\MR\08\361

高级

趣味指数：★★★★

实例说明

删除图书信息表中的所有数据或者说清空数据表的方法在实际编程中是很少见的，一般都是删除指定的一条或几条数据。本实例通过 delete 语句的子句实现删除图书信息表中的指定数据，运行结果如图 8.28 所示。

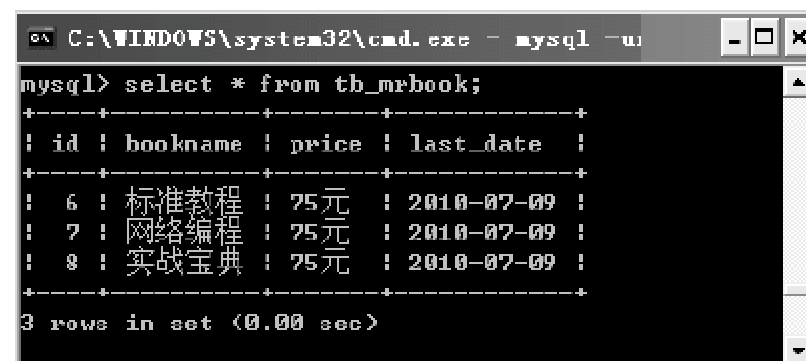


图 8.28　指定删除 id 等于 5 的数据

关键技术

本实例的关键点是 where 条件句的灵活运用。where 子句用来选取需要检索的记录。因为一个表通常会有数千条记录，在查询结果中，用户仅需其中的一部分记录，这时就需要使用 Where 子句指定一系列的查询条件。下面是 where 子句最简单的语法：

```
SELECT<字段列表>
FROM<表名>
WHERE<条件表达式>
```

为了满足多种不同的查询，where 子句提供了丰富的搜索条件，下面将给出 where 子句常用的比较运算符，如表 8.1 所示。

表 8.1　where 子句常用的比较运算符

运　算　符	说　　明
=	等于
>	大于
<	小于
>=	大于等于
<=	小于等于
!>	不大于
!<	不小于
<>或!=	不等于

设计过程

（1）开启 MySQL 服务，输入命令连接 MySQL 服务器，创建数据库和数据表，将数据表命名为 tb_mrbook，并通过 insert 语句向数据表里添加信息，通过 select 语句查看表信息。

（2）通过 delete 语句删除 id 等于 5 的数据，运行结果如图 8.29 所示。

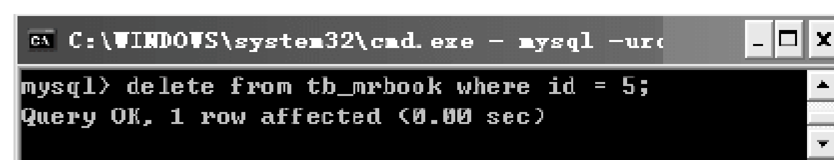


图 8.29　删除成功提示

秘笈心法

心法领悟 361：where 子句的优点。

where 子句是非常灵活的，其使用方法与 PHP 中 if()语句有些类似，也可以用于执行比较或逻辑运算。

8.5　PHP 操作 MySQL 数据库

PHP 支持的数据库类型较多，在这些数据库中，MySQL 数据库与 PHP 结合最好，与 Linux 系统、Apache 服务器和 PHP 语言构成当今主流的 LAMP 网站架构模式，并且 PHP 提供多种操作 MySQL 数据库的方式，因而适合不同需求和不同类型项目的需要。

实例 362	mysql_connect()函数连接 MySQL 服务器 光盘位置：光盘\MR\08\362	高级 趣味指数：★★★★

实例说明

通过 PHP 语言连接 MySQL 数据库与在 cmd 命令提示符下操作数据库有所不同。例如，设置页面的编码格式为 GBK，在 MySQL 命令提示符下只需输入 set names gbk 即可，而在 PHP 中需使用函数 mysql_query(“set names GBK”)实现。所以用户在学习操作 MySQL 数据库的同时还要与 PHP 函数联合应用，下面就为用户讲解使用 PHP 操作 MySQL 数据库。本实例通过 mysql_connect()函数连接 MySQL 服务器，运行结果如图 8.30 所示。



图 8.30　连接 MySQL 服务器

关键技术

使用 mysql_connect()函数打开一个到 MySQL 服务器的连接，其语法如下：

```
resource mysql_connect ( [string server [, string username [, string password [, bool new_link [, int client_flags]]]]] );
```

参数说明：

string server：表示连接数据库的 HOST。

string username：表示 MySQL 数据库用户名。

string password：表示 MySQL 数据库密码。

设计过程

新建 index.php 文件，创建 form 表单提交连接 MySQL 服务器的用户名、密码和服务器。获取 form 表单提交的数据，应用 mysql_connect()函数完成与 MySQL 服务器的连接，其代码如下：

```
<?php
if($_POST[sub]){                                                    //单击“确定”按钮
    if($_POST[na] == "" || $_POST[nam] == "" || $_POST[name] == ""){   //检测文本框是否为空
        echo "<script>alert('文本框不能为空');</script>";                //输出提示
    }else{
        if(@mysql_connect("$_POST[na]","$_POST[nam]","$_POST[name]")){  //连接 MySQL 服务器
            echo "<script>alert('连接 MySQL 服务器成功');</script>";     //输出提示
        }else{
```

```
                        echo "<script>alert('错误了');</script>";
                }
        }
}
?>
```

秘笈心法

心法领悟 362：连接 MySQL 服务器。

安装 MySQL 数据库时，会提示用户设置 Host、用户名和密码。如果用户并未进行设置，MySQL 在默认安装的情况下，Host 为 localhost、用户名为 root、密码为 NULL（空）。

实例 363	mysql_select_db()函数选择 MySQL 数据库 光盘位置：光盘\MR\08\363	高级 趣味指数：★★★★

实例说明

通过函数 mysql_connect()可以与 MySQL 服务器建立连接，在 cmd 命令提示符下使用 use 语句选择想要操作的数据库。在 PHP 代码中需要使用函数 mysql_select_db()选择并连接数据库。本实例通过函数 mysql_select_db()实现与 MySQL 数据库的选择与连接，运行结果如图 8.31 所示。



图 8.31　使用函数选择并连接 MySQL 数据库

关键技术

mysql_select_db()函数，用于选择 MySQL 数据库。

```
bool mysql_select_db ( string database_name [, resource link_identifier] );
```

参数说明：

string database_name：表示选择 MySQL 数据库名称。

resource link_identifier：表示 MySQL 连接标识。

设计过程

（1）新建 index.php 文件。首先连接 MySQL 服务器，通过 mysql 函数库的函数读取输出 MySQL 数据库中存储的数据库名称，并且通过下拉列表框显示所有数据库名称，代码如下：

```
<?php
        $conn = mysql_connect("localhost","root","111");                    //连接 MySQL 服务器
        $rs = mysql_query("show databases");                                //查询操作
        while($rst = mysql_fetch_array($rs)){                               //while 循环
                $a = 0;
                echo "<option value=".$rst[$a].">".$rst[$a]."</option><br>";        //输出数据库名称
                $a++;
        }
?>
```

（2）当单击“确定”按钮时，将下拉列表框的选择信息以参数的形式传递给函数 mysql_select_db()，并输出提示信息，其代码如下：

```
<?php
    if($_POST[sub]){                                                    //通过 POST 方式传递参数
        if($_POST[select] == "1"){                                      //当下拉列表框值为 1 时
            echo "<script>alert('请选择数据库');</script>";              //输出提示信息
        }else{
            if(mysql_select_db("$_POST[select]",$conn)){                //如果不为 1，连接数据库
                mysql_close();                                          //关闭连接
                echo "<script>alert('已选择指定数据库');</script>";      //输出提示
            }else{
                echo "<script>alert('出错误了');</script>";              //提示信息
            }
        }
    }
?>
```

秘笈心法

心法领悟 363：两种方法判断数据库是否连接成功。

本实例是利用函数 mysql_select_db()的返回值，判断是否连接到数据库从而输出提示信息。还可以通过函数 mysql_select_db() or die()来判断是否连接到数据库，如果连接到数据库将不会有任何信息提示，否则输出错误信息。

实例 364	mysql_query()函数执行 SQL 语句 光盘位置：光盘\MR\08\364	高级 趣味指数：★★★★

实例说明

用 PHP 代码操作 MySQL 数据库的主要目的是为了让程序控制数据库信息的增删改查等操作，而函数 mysql_query()可以看作是部分 SQL 语句的载体。本实例通过 mysql_query()函数执行 SQL 语句，运行结果如图 8.32 所示。

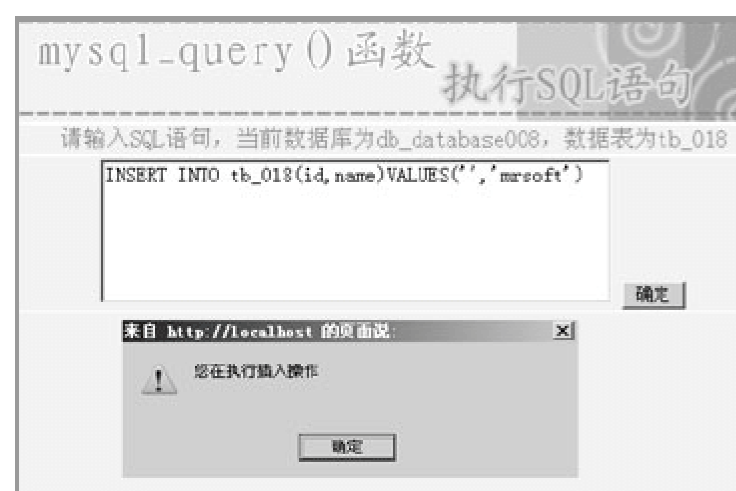


图 8.32　mysql_query()函数执行 SQL 语句

关键技术

使用 mysql_query()函数可向与指定的连接标识符关联的服务器中当前活动的数据库发送一条 MySQL 查询，其语法如下：

```
resource mysql_query ( string query [, resource link_identifier] );
```

参数说明：

resource link_identifier：连接标识。

string query：SQL 语句。

设计过程

新建 index.php 文件。首先连接服务器，连接 db_database08 数据库。然后，获取 form 表单中提交的 SQL

语句，并且对 SQL 语句的类型进行判断，根据不同的类型应用 mysql_query()函数执行不同的操作，其关键代码如下：

```
if($_POST[sub]){                                                          //单击按钮
 $sql = $_POST[text];                                                     //定义变量
 if($sql == ""){                                                          //如果文本框为空
        echo "<script>alert('您输入的文本框内容为空');</script>";              //JavaScript 提示
        }else{
               $conn = mysql_connect("localhost","root","111") or die("连接 MySQL 出现错误");   //连接 MySQL
               mysql_select_db("db_database08",$conn) or die("false");              //连接数据库
                      $sql1 = substr($sql,0,6);                                   //截取 SQL 语句前 6 个字符
                      if(strtoupper($sql1) == "SELECT"){
                             if(mysql_query($sql)){                               //执行查询操作
                                    echo "<script>alert('您在执行查询操作');</script>";
                             }else{
                                    echo "<script>alert('SQL 语句出现错误');</script>";
                             }
                      }else{
                             if(strtoupper($sql1) == "INSERT"){
                                    if(mysql_query($sql)){
                                           echo "<script>alert('您在执行插入操作');</script>";
                                    }else{
                                           echo "<script>alert('SQL 语句出现错误');</script>";
                                    }
                             }else{
                                    if(strtoupper($sql1) == "UPDATE"){
                                           if(mysql_query($sql)){
                                                  echo "<script>alert('您在更新查询操作');</script>";
                                           }else{
                                                  echo "<script>alert('SQL 语句出现错误');</script>";
                                           }
                                    }
                             }
                      }
               }
 }
?>
```

在对 SQL 语句的类型进行判断时，首先应用 substr()函数截取 SQL 语句的前 6 个字节，然后将截取的内容转换为小写，并通过这 6 个字符串对 SQL 语句的类型进行判断。

■ 秘笈心法

心法领悟 364：mysql_query()中连接标识符的作用。

mysql_query() 向与指定的连接标识符关联的服务器中当前活动的数据库发送一条查询。如果没有指定 link_identifier，则使用上一个打开的连接，如果没有打开的连接，本函数会尝试无参数调用 mysql_connect() 函数来建立一个连接并使用，其查询结果会被缓存。.

实例 365	mysql_fetch_array()函数将结果返回到数组中 光盘位置：光盘\MR\08\365	高级 趣味指数：★★★★

■ 实例说明

用户在进行查询操作时，目的是为了将结果显示到网页中，下面介绍如何通过 mysql_fetch_array()函数将查询结果返回到数组中。本实例通过函数 mysql_fetch_array()将查询结果返回到数组中，打印数组的结果如图 8.33 所示，循环输出数组中数据的结果如图 8.34 所示。

mysql_fetch_array()函数 将结果返回到数组中

数组输出 表格输出

Array ([0] => 1 [id] => 1 [1] => 明日科技 [name] => 明日科技 [2] => mrsoft [pwd] => mrsoft [3] => 2010-07-13 [last_date] => 2010-07-13)

Array ([0] => 2 [id] => 2 [1] => 小杨 [name] => 小杨 [2] => 1081 [pwd] => 1081 [3] => 2010-07-13 [last_date] => 2010-07-13)

Array ([0] => 3 [id] => 3 [1] => 小潘 [name] => 小潘 [2] => 1026 [pwd] => 1026 [3] => 2010-07-13 [last_date] => 2010-07-13)

Array ([0] => 4 [id] => 4 [1] => 小刘 [name] => 小刘 [2] => 1015 [pwd] => 1015 [3] => 2010-07-13 [last_date] => 2010-07-13)

图 8.33 打印数组

mysql_fetch_array()函数 将结果返回到数组中

数组输出 表格输出

ID	用户名	密码	时间
1	明日科技	mrsoft	2010-07-13
2	小杨	1081	2010-07-13
3	小潘	1026	2010-07-13
4	小刘	1015	2010-07-13

图 8.34 打印表格

关键技术

mysql_fetch_array()函数用于从结果集中取得一行作为关联数组或数字数组，或二者兼有，其语法如下：

```
array mysql_fetch_array ( resource result [, int result_type] );
```

参数 resource result 为结果集。

设计过程

（1）创建 PHP 脚本文件，命名为 in.php，将 MySQL 数据库的连接操作封装到类中，定义构造函数为成员变量赋值，定义 connect()函数实现与 MySQL 服务器和数据库的连接。MySQL 类的代码如下：

```
<?php
class mysql{
	var $conn ;
	var $host;
	var $username;
	var $pwd;
	var $db;
	function __construct($host,$username,$pwd,$db){                    //为成员变量赋值
		$this -> host = $host;
		$this -> username = $username;
		$this -> pwd = $pwd;
		$this -> db = $db;
		$this -> connect();
	}
	function connect(){
		$this -> conn = $conn;                                          //定义连接标识
		$this -> conn = mysql_connect($this -> host,$this -> username,$this -> pwd);   //连接数据库服务器
		mysql_select_db($this -> db,$this -> conn);                     //连接数据库
		mysql_query("SET NAMES UTF8");                                  //设置数据库编码
	}
}
?>
```

（2）创建 PHP 脚本文件，命名为 index.php。首先，包含数据库连接类文件。然后，执行类的实例化操作，完成与数据库的连接，并返回连接对象。最后，定义 SQL 查询语句，通过 mysql_query()函数执行查询操作，通过 mysql_fetch_array()函数将查询结果集存储到数组中，并通过 while 语句循环输出查询结果集中的数据，其代码如下：

```
<?php
if($_GET[arra] == "1"){                                                 //如果地址栏参数 arra 等于 1
	include("in.php");                                                  //包含数据库连接类
	$my = new mysql("localhost","root","111","db_database08");          //实例化数据库连接类
	$sql = "select * from tb_020";                                      //sql 语句
	$rs = mysql_query($sql);                                            //执行查询操作并返回结果集
	while($rst = mysql_fetch_array($rs)){                               //循环数组
		print_r($rst);                                                  //打印数组
	}
}
if($_GET[table] == "1"){                                                //如果地址栏参数 table 等于 1
```

```
        include("in.php");                                                  //包含数据库连接类
        $my = new mysql("localhost","root","111","db_database08");          //实例化对象
        $sql = "select * from tb_020";                                      //sql 语句
        $rs = mysql_query($sql);                                            //返回结果集
?>
        <table border="1" bordercolor="#FFCC00" cellspacing="0px" width="580px">
        <tr><td>ID</td><td>用户名</td><td>密码</td><td>时间</td></tr>
<?php
        while($rst = mysql_fetch_array($rs)){                               //循环数组
                echo "<tr>";
                echo "<td>".$rst[0]."</td><td>".$rst[1]."</td><td>".$rst[2]."</td><td>".$rst[3]."</td>";  //表格式输出
                echo "</tr>";
        }
?>
        </table>
<?php
}
?>
```

秘笈心法

心法领悟 365：使用 mysql_fetch_array()函数的注意事项。

在使用 mysql_fetch_array()函数时，如果结果中的两个或以上的列具有相同字段名，最后一列将优先。要访问同名的其他列，必须用该列的数字索引或给该列起个别名，对有别名的列，不能再用原来的列名访问其内容。

实例 366　mysql_fetch_row()函数从结果集中获取数据　高级

光盘位置：光盘\MR\08\366　趣味指数：★★★★

实例说明

获取查询结果集中数据信息的函数很多，下面推荐一个新的获取结果集中数据的函数——mysql_fetch_row()。本实例的运行结果如图 8.35 所示。

mysql_fetch_row()函数
从结果集中获取数据

查 询　下一条

ID	用戶名	密碼	時間
1	小楊	1081	2010-07-14

图 8.35　查询一条数据

关键技术

mysql_fetch_row()函数用于从结果集中获取一行作为枚举数组，返回根据所获取的行生成的数组，如果没有更多行则返回 FALSE，其语法如下：

```
array mysql_fetch_row ( resource result );
```

参数 result 为指定的结果集。

设计过程

（1）新建 in.php 文件，封装数据库的连接类。

（2）新建 index.php 文件。首先，包含数据库连接类文件。然后，执行类的实例化操作，完成与数据库的连接，并返回连接对象。最后，通过 mysql_fetch_row()函数将查询结果集存储到数组中，并输出获取到的数据，其代码如下：

```
<?php
include ("in.php");
$my = new mysql ( "localhost", "root", "111", "db  database08" );
if ($_GET [id] == 1) {
        $sql = "select * from tb_demo021";
        $rs = mysql  query ( $sql );
        $rst = mysql_fetch_row ( $rs );
?>
<?php
        echo "<tr><td class='one'>" . $rst [0] . "</td><td class='one'>" . $rst [1] . "</td><td class='one'>" . $rst [2] . "</td><td class='one'>" . $rst [3] .
"</td></tr>";
}
//单击“下一条”按钮，类似分页
if ($_GET [id] == 2) {
        $os = mysql_query ( "select * from tb_demo021_order order by id desc limit 1" );
        $ost = mysql  fetch  row ( $os );
        $id = $ost [1];
        $rs = mysql_query ( "select * from tb_demo021 limit $id,1" );
        $id = $id + 1;
        $rst = mysql_fetch_row ( $rs );
?>
```

秘笈心法

心法领悟 366：mysql_fetch_row()函数的特点。

mysql_fetch_row()函数是从指定的结果标识关联的结果集中取得一行数据并作为数组返回，每个结果的列储存在一个数组的单元中，偏移量从 0 开始。依次调用 mysql_fetch_row()函数将返回结果集中的下一行，如果没有更多行则返回 FALSE。

实例 367 mysql_num_rows()函数获取结果集中记录数

光盘位置：光盘\MR\08\367

高级

趣味指数：★★★★

实例说明

本实例介绍如何通过 PHP 的函数 mysql_num_rows()统计数据库中的记录数，运行结果如图 8.36 所示。

mysql_num_rows()函数 获取结果集中记录数

获取数据库 获取数据表 获取信息

ID	用户	密码	时间
1	小杨	1081	2010-07-14
2	小潘	1026	2010-07-14
3	小刘	1018	2010-07-14

当前共有数据表3个

图 8.36 tb_demo021 下的信息数量

关键技术

mysql_num_rows()函数，用于取得结果集中行的数目。

```
int mysql_num_rows ( resource result );
```

参数 result 指定操作的结果集，其返回值为 int 类型。

设计过程

（1）新建 in.php 文件，封装数据库的连接类。

（2）新建 index.php 文件。首先，包含数据库连接类文件。然后，执行类的实例化操作，完成与数据库的

连接，并返回连接对象。最后，通过 mysql_num_rows()函数统计数据库、数据表的个数以及指定数据表的总记录数，其关键代码如下：

```
<?php
include ("in.php");
$my = new mysql ( "localhost", "root", "111", "db_database08" );
if ($_GET [id] == 1) {
        $rs = mysql_query ( "show databases" );
        $num = mysql_num_rows ( $rs );
        while ( $rst = mysql_fetch_row ( $rs ) ) {
                print_r ( $rst );
}
        echo "<br><b class='two'>当前共有数据库<b class='three'>" . $num . "</b>个</b>";
}
```

上述代码查询 MySQL 数据库中包含的数据库，通过 mysql_num_rows()函数统计数据库的个数，应用 while 语句循环输出数据库的名称。

秘笈心法

心法领悟 367：使用 mysql_num_rows()函数的注意事项。

mysql_num_rows()函数仅对 select 查询语句有效。

实例 368　mysql_result()函数获取结果数据

光盘位置：光盘\MR\08\368

高级

趣味指数：★★★★

实例说明

如果想要查看数据表中某个单元的数据，那么推荐使用 mysql_result()函数。本实例将介绍如何使用 mysql_result()函数获取结果集中指定单元的数据，运行结果如图 8.37 所示。

mysql_result()函数 获取结果数据

横向：用户　纵向：2　确定

取得字段为小潘

ID	用户	密码	日期
1	小杨	1081	2010-07-14
2	小潘	1026	2010-07-14
3	小刘	1015	2010-07-14
4	明日	mrsoft	2010-07-14

图 8.37　取得一条字段

关键技术

mysql_result()函数，用于返回 MySQL 结果集中一个单元的内容。

```
mixed mysql_result ( resource result, int row [, mixed field] );
```

参数说明：

resource result：结果集。

int row：行数。

mixed field：可选参数，字段名称。

设计过程

（1）创建脚本文件，命名为 in.php，利用面向对象知识编写数据库连接类。

（2）创建脚本文件，命名为 index.php。首先，包含数据库连接类，对类进行实例化，连接数据库。然后，通过 while 语句和 mysql_fetch_row()函数循环输出数据表中数据。最后，根据 form 表单提交的数据进行判断，在 switch 语句中通过 mysql_result()函数获取数据表中指定单元的数据，其关键代码如下：

```
<?php
        }
        if($_POST[sub]){                                       //判断提交按钮的值是否为真
            if($_POST[name] == 1 || $_POST[name] < 1){
                echo "<script>alert('选择的是表格表头');</script>";
            }else{
            switch($_POST[na]){                                //在 switch 语句中，根据表单提交的值进行判断
                case "ID";
                $rse = mysql_result($rs,$_POST[name]-1,"id");   //获取指定单元的数据
                echo "取得字段为".$rse;
                break;
                case "用户";
                $rse = mysql_result($rs,$_POST[name]-1,"name");
                echo "取得字段为".$rse;
                break;
                case "密码";
                $rse = mysql_result($rs,$_POST[name]-1,"pwd");
                echo "取得字段为".$rse;
                break;
                case "日期";
                $rse = mysql_result($rs,$_POST[name]-1,"date");
                echo "取得字段为".$rse;
                break;
            }
        }
}
?>
```

秘笈心法

心法领悟 368：使用 mysql_result()函数的注意事项。

当 mysql_result()函数作用于很大的结果集时，应该考虑使用能够取得整行的函数。这些函数在一次函数调用中返回多个单元的内容，效率高于 mysql_result()函数。

实例 369 Insert 语句添加图书信息

光盘位置：光盘\MR\08\369

高级

趣味指数：★★★★

实例说明

MySQL 数据库的主要作用在于能够存储大量的数据，作为数据的规范化载体，可以通过 Insert 添加语句将数据添加到数据表中。本实例通过 INSERT 语句向图书信息表中添加信息，运行结果如图 8.38 所示。

图 8.38　插入数据

关键技术

insert 语句：向数据库插入一条数据，语法如下：

```
INSERT [LOW_PRIORITY | DELAYED | HIGH_PRIORITY] [IGNORE]
    [INTO] tbl_name [(col_name,...)]
    VALUES ({expr | DEFAULT},...),(...),...
    [ ON DUPLICATE KEY UPDATE col_name=expr, ... ]
```

本实例中实现添加图书信息的 SQL 语句如下：

```
$sql = "insert into tb_demo024(id,name,pwd,date)values('','$_POST[n]','$_POST[na]','$_POST[nam]')";
```

设计过程

（1）建立 in.php 文件，编写 MySQL 数据库的连接类。

（2）编写 index.php 文件，创建 form 表单，提交要添加的图书信息，将数据提交到本页。获取表单中提交的数据，连接数据库，通过 insert 语句将表单提交的数据添加到指定的数据表中，最终循环输出数据表中的数据，其关键代码如下：

```
if($_POST[sub2]){
    if($_POST[n] != "" || $_POST[na] != "" || $_POST[nam] != ""){           //判断提交的数据是否为空
        if(preg_match("/\d/",$_POST[na])){                                  //通过正则表达式判断提交数据格式是否正确
            $sql = "insert into tb_demo024(id,name,pwd,date)values('','$_POST[n]','$_POST[na]','$_POST[nam]')";
            if(mysql_query($sql)){                                          //执行添加语句
                echo "<script>alert('插入数据成功');location.href='index.php?id=3';</script>";
            }
        }else{
            echo "<script>alert('您输入的不是一个数字');</script>";
        }
    }else{
        echo "<script>alert('文本框为空');</script>";
    }
}
```

秘笈心法

心法领悟 369：使用 insert 语句的注意事项。

在 PHP 中，通过 insert 语句向数据表中添加数据时，要做到三点统一：

（1）form 表单中定义的表单元素名称与在数据处理页中通过$_POST[]方法获取表单元素值使用的名称统一，同时还要注意它们是区分字符大小写的。如果两个名称不统一，那么将获取不到指定的值。

（2）在编辑的 insert 添加语句中，字段的名称与字段的值必须对应，不能将 varchar 类型的数据添加到 date 类型的字段中。

（3）在编辑的 insert 添加语句中，字段的名称必须与数据表中定义的字段名称统一，否则同样会导致添加失败。

实例 370	Select 语句查询图书信息 光盘位置：光盘\MR\08\370	高级 趣味指数：★★★★

实例说明

数据库可以与 PHP 页面进行很好的交互，既可以向数据库中插入数据，也同样可以从数据库中查询数据。本实例通过 SELECT 语句实现查询图书信息，运行结果如图 8.39 所示。

SELECT语句查询图书信息

查询图书信息

ID	书名	价格	日期
1	《PHP开发实战宝典》	75元	2010-07-15
2	《PHP网络编程自学手册》	75元	2010-07-15
3	《PHP标准教程》	75元	2010-07-15
4	《PHP范例宝典》	75元	2010-07-15

图 8.39　查询图书信息

关键技术

SELECT 语句是在 SQL 中比较常用的语句，也是比较重要的语句。使用 SELECT 语句可以从数据表或视图中进行查询，并将查询结果以表格的形式返回，以表格返回的结果也可以称为结果集。SELECT 语句中的主要结构如下：

```
SELECT    select_list
[ INTO new  table  name]
FROM table_list
[ WHERE search_conditions]
[ GROUP BY group  by  list]
[ HAVING search_conditions]
[ ORDER BY order_list [ASC| DESC ]]
```

参数说明：

select_list：指定需要查询返回的列。多个列表之间使用逗号分隔。在选择列表时也可以使用“*”符号来表示返回表中的所有列。

INTO new_table_name：创建新表并将查询行插入新表中。new_table_name 指定新表的名称。

FROM table_list：指定需要查询的表。这些来源表可能包括基表、视图和链接表。FROM 子句还可包含连接说明，该说明定义了 SQL Server 用来在表之间进行导航的特定路径。

WHERE search_conditions：WHERE 子句指定用于限制返回的行的搜索条件。

GROUP BY group_by_list：GROUP BY 子句根据 group_by_list 列中的值将结果集分成组。例如，student 表在“性别”中有两个值。GROUP BY ShipVia 子句将结果集分成两组，每组对应于 ShipVia 的一个值。

HAVING search_conditions：HAVING 子句指定组或聚合的搜索条件。逻辑上讲，HAVING 子句从中间结果集对行进行筛选，这些中间结果集是用 SELECT 语句中的 FROM、WHERE 或 GROUP BY 子句创建的。HAVING 子句通常与 GROUP BY 子句一起使用，尽管 HAVING 子句前面不必有 GROUP BY 子句。

ORDER BY order_list [ASC | DESC]：ORDER BY 子句定义结果集中的行排列的顺序。order_list 指定组成排序列表的结果集的列。ASC 和 DESC 关键字用于指定行是按升序还是按降序排序。ORDER BY 之所以重要，是因为其关系理论规定除非已经指定 ORDER BY，否则不能假设结果集中的行带有任何序列。如果结果集中行的顺序对于 SELECT 语句来说很重要，那么在该语句中就必须使用 ORDER BY 子句。

设计过程

（1）建立 in.php 文件，通过面向对象的知识编写数据库连接类。

（2）编写 index.php 文件，包含数据库连接类，实例化对象，连接数据库。然后，执行查询语句，通过 while 语句和 mysql_fetch_row()函数循环输出查询结果，核心代码如下：

```
<?php
include("in.php");                                                    //包含数据库连接类
new mysql("localhost","root","111","db_database08");                  //实例化对象
if($_GET[id] == 1){                                                   //如果地址栏参数 id 等于 1
        $rs = mysql_query("select * from tb_demo025");                //返回结果集
        echo "<table width='580px'>";                                 //输出结果
        echo "<tr><td    style='color:#FFFFFF'>ID</td><td bgcolor='#FFFFFF'>书名</td><td style='color:#FFFFFF'>价格</td><td
bgcolor='#FFFFFF'>日期</td></tr>";
        while($rst = mysql_fetch_row($rs)){
```

```
            echo "<tr><td   >".$rst[0]."</td><td bgcolor='#FFFFFF'>".$rst[1]."</td><td >".$rst[2]."</td><td>".$rst[3]."</td></tr>";
        }
        echo "</table>";
}
?>
```

秘笈心法

心法领悟 370：SELECT 语句的优点。

灵活运用 SELECT 语句，可以编写出非常复杂的查询，如单列查询、多列查询、查询所有列、别名的应用、TOP 查询前若干行、计算列查询、条件查询和范围查询等。

实例 371　分页显示图书信息

光盘位置：光盘\MR\08\371

高级

趣味指数：★★★★

实例说明

select 语句可以查询数据库信息，但是如果数据库信息非常多，在一个页面中显示所有数据，首先会使页面显得繁杂、不美观，其次会给用户造成网站不够智能的表象。本实例灵活运用 SELECT 语句实现对图书信息的分页处理，其运行结果如图 8.40 所示。

图 8.40　分页显示图书信息

关键技术

本实例应用 select 查询语句及其子句 limit，实现图书信息的分页显示。通过 limit 关键字控制当前页面显示的记录数和开始位置。本实例分页查询图书信息的 SQL 语句代码如下：

```
$sql = "select * from tb_demo026 limit ".(($page - 1)*$num).",".$num;
```

其中(($page - 1)*$num)计算的是当前页的开始位置，而$num 是当前页显示的记录数。

设计过程

（1）建立 in.php 文件，通过面向对象的知识编写数据库连接类。

（2）编写 index.php 文件。首先，包含数据库连接类并实例化对象。然后，定义查询语句统计查询结果集中的记录数，定义分页变量。接着，定义分页查询语句，每页显示 5 条记录。最后，创建分页超链接，完成图书信息的分页显示，核心代码如下：

```
<?php
include("in.php");                                              //包含数据库连接类
new mysql("localhost","root","111","db_database08");            //实例化对象
if($_GET[id] == 1){                                             //如果地址栏参数等于 1
        $num = 5;                                               //定义变量
        $rse = mysql_query("select * from tb_demo026");         //执行查询操作
        $nu = mysql_num_rows($rse);                             //返回数据库信息条数
        if(!$_GET[page]){                                       //设置地址栏参数 page
                $page = 1;
```

```
        }else{
                $page = $_GET[page];
        }
        $sql = "select * from tb_demo026 limit ".(($page - 1)*$num).",".$num;//分页 sql 语句
        $rs = mysql_query($sql);                                          //返回结果集
        $nur = ceil($nu/$num);                                            //取整函数
        echo "<a href='index.php?id=1&page=1'>首页</a>";                   //设置主页地址栏参数
        if($page >= 2){                                                   //如果地址栏参数 page 值大于等于 2
?>
                <a href="index.php?id=1&page=<?php echo $page-1;?>">上一页</a>      //上一页
<?php
        }
        if($page < $nur){                                                 //如果 page 小于总页数
?>
                <a href="index.php?id=1&page=<?php echo $page+1;?>">下一页</a>      //下一页
<?php
        }
?>
                <a href="index.php?id=1&page=<?php echo $nur;?>">尾页</a>          //尾页
<?php
        echo "<table width='580px'><tr><td width='100px' bgcolor='#FFFFFF'>ID</td><td width='200px'>书名</td><td width='180px'
bgcolor='#FFFFFF'>价格</td><td width='100px'>日期</td></tr>";
        while($rst = mysql_fetch_row($rs)){
                echo "<tr><td bgcolor='#FFFFFF'>".$rst[0]."</td><td>《".$rst[1]."》</td><td bgcolor='#FFFFFF'>".$rst[2]."</td><td>".$rst[3]."</td></tr>";
        }
        echo "</table>";
}
?>
```

秘笈心法

心法领悟 371：　SELECT 语句的子句 limit。

Select 语句的子句 limit 功能很强大，可用于控制查询语句的开始位置以及查询的数量。但是，limit 子句是 MySQL 数据库特有的，在 SQL Server 或者 Access 数据库中是不支持的。

实例 372	Update 语句更新图书信息 光盘位置：光盘\MR\08\372	高级 趣味指数：★★★★

实例说明

程序员在手动插入大量数据的时候，难免将图书信息的某个字段书写错误，这个时候就需要使用 update 语句更新信息。本实例通过 update 语句实现更新图书信息，运行结果如图 8.41 所示。

图 8.41　更新图书信息

关键技术

update 语句用于修改表中的数据，通常有两种形式，一是可以一次向表中修改多条数据（实现一改全改）；

二是只修改一条记录，其语法格式如下：

```
UPDATE{
        table_name WITH ( < table_hint_limited > [ ...n ] )
        | view  name
        | rowset_function_limited
        }
           SET
             { column_name = { expression | DEFAULT | NULL }
             | @variable = expression
             | @variable = column = expression } [ ,...n ]
      { { [ FROM { < table_source > } [ ,...n ] ]
             [ WHERE
                   < search  condition > ] }
             |
             [ WHERE CURRENT OF
             { { [ GLOBAL ] cursor  name } | cursor  variable  name }
             ] }
             [ OPTION ( < query_hint > [ ,...n ] ) ]
< table  source > ::=
      table_name [ [ AS ] table_alias ] [ WITH ( < table_hint > [ ,...n ] ) ]
      | view_name [ [ AS ] table_alias]
      | rowset  function [ [AS] table  alias]
      | derived_table [AS] table_alias [ ( column_alias [ ,...n ] ) ]
      | < joined_table >
< joined  table > ::=
      < table_source > < join_type > < table_source > ON < search_condition >
      | < table_source > CROSS JOIN < table_source >
      | < joined  table >
< join_type > ::=
      [ INNER | { { LEFT | RIGHT | FULL } [OUTER] } ]
      [ < join_hint > ]
      JOIN
< table_hint_limited > ::=
      {      FASTFIRSTROW
             | HOLDLOCK
             | PAGLOCK
             | READCOMMITTED
             | REPEATABLEREAD
             | ROWLOCK
             | SERIALIZABLE
             | TABLOCK
             | TABLOCKX
             | UPDLOCK
      }
< table_hint > ::=
      {      INDEX ( index_val [ ,...n ] )
             | FASTFIRSTROW
             | HOLDLOCK
             | NOLOCK
             | PAGLOCK
             | READCOMMITTED
             | READPAST
             | READUNCOMMITTED
             | REPEATABLEREAD
             | ROWLOCK
             | SERIALIZABLE
             | TABLOCK
             | TABLOCKX
             | UPDLOCK
      }
< query_hint > ::=
      {      { HASH | ORDER } GROUP
             | { CONCAT | HASH | MERGE } UNION
             | {LOOP | MERGE | HASH } JOIN
             | FAST number  rows
             | FORCE ORDER
             | MAXDOP
             | ROBUST PLAN
             | KEEP PLAN
```

```
    }
```

参数说明：

table_name：要更新的表的名称。如果该表不在当前服务器或数据库中，或不为当前用户所有，此名称可用连接服务器、数据库和所有者名称来限定。

WITH(<table_hint_limited>[...n])：指定目标表所允许的一个或多个表提示。需要有 WITH 关键字和圆括号，不允许有 READPAST、NOLOCK 和 READUNCOMMITTED。

view_name：要更新的视图的名称。通过 view_name 来引用的视图必须是可更新的。用 update 语句进行的修改，至多只能影响视图的 FROM 子句所引用的基表中的一个。

rowset_function_limited：OPENQUERY 或 OPENROWSET 函数，视提供程序功能而定。

SET：指定要更新的列或变量名称的列表。

column_name：含有要更改数据的列的名称。column_name 必须驻留于 update 子句所指定的表或视图中，标识列不能进行更新。

设计过程

（1）建立 in.php 文件，通过面向对象的知识编写数据库连接类。

（2）编写 index.php 文件。首先，包含数据库连接文件，执行类的实例化并连接数据库。然后，循环输出数据表中数据，并且为每条记录创建一个“更新”超链接，链接到当前页，传递参数为该条记录的 ID 值。最后，根据超链接传递的参数值进行判断，如果符合要求则输出 form 表单，将更新数据提交到本页，完成数据的更新操作，核心代码如下：

```
<?php
include("in.php");
new mysql("localhost","root","111","db_database08");                                    //连接数据库
if($_POST[sub1]){
        if($_POST[na] == "更新书名" || $_POST[nam] == "更新价格" || $_POST[name] == "更新日期"){
                echo "<script>alert('有文本框未做更新，请重新填写');location.href='index.php?id=1'</script>";
        }else{
                $sql = "update tb_demo027 set bname='$_POST[na]',price='$_POST[nam]',date='$_POST[name]' where id='$_GET[update]'";
                if(mysql_query($sql)){
                        echo "<script>alert('更新成功');</script>";
                }
        }
}
?>
```

秘笈心法

心法领悟 372：使用 update 语句的注意事项。

在使用 update 更新语句时，要注意 where 子句的使用，如果不存在 where 子句，则会更新数据库中的所有数据。所以，在应用 update 语句执行更新操作时，必须指定更新的条件，而这个条件最理想的选择就是数据的 ID，即数据表中字段的主键。

实例 373 图书信息的批量更新

光盘位置：光盘\MR\08\373

高级

趣味指数：★★★★

实例说明

对于数据类型相同的数据信息，用户没有必要对数据进行逐条更新，可以通过批量更新一次解决。本实例通过 Update 更新语句实现图书信息的批量更新，运行结果如图 8.42 所示。

ID	书名	价格	日期
1	PHP开发实战宝典	80元	2010-07-17
2	VB开发实战宝典	80元	2010-07-17
3	JAVA开发实战宝典	79元	2010-07-17
4	JAVAWEB开发实战宝典	78元	2010-07-17
5	.NET开发实战宝典	78元	2010-07-17
6	C#开发实战宝典	78元	2010-07-17

图 8.42　批量更新

■ 关键技术

本实例实现批量更新图书信息的 SQL 语句的代码如下：

```
update tb_demo028 set date='$_POST[name]'where (id>=$_POST[na] and id<=$_POST[nam]);
```

本实例的匹配更新是通过 where 子句的执行条件来完成的，更新的是 date 字段的值，其更新的范围是 id 大于$_POST[na]、小于等于$_POST[nam]的数据。

■ 设计过程

（1）建立 in.php 文件，通过面向对象的知识编写数据库连接类。

（2）编写 index.php 文件。首先，创建 form 表单，将更新数据提交到本页。然后，包含数据库连接类并实例化对象。接着，判断表单中提交的数据，根据判断结果执行不同的更新语句。最后，循环输出数据表中数据，核心代码如下：

```
<?php
include("in.php");
new mysql("localhost","root","111","db_database08");                    //连接数据库
if($_POST[sub]){
    if($_POST[n] == "填入首字段名称" || $_POST[name] == "更新内容" || $_POST[na] == "首字母 ID" || $_POST[nam] == "尾字母 ID"){
        echo "<script>alert('请将文本框内容填写完整');</script>";
    }
    if($_POST[n] == "ID" || $_POST[n] == "id" || $_POST[n] == "Id" || $_POST[n] == "iD"){
        echo "<script>alert('ID 主键无法更新');</script>";
    }
    if($_POST[n] == "书名"){
        if(mysql_query("update tb_demo028 set bname='$_POST[name]'where (id>=$_POST[na] and id<=$_POST[nam])")){
            echo "<script>alert('批量更新书名完成');location.href='index.php'</script>";
        }else{
            echo "<script>alert('输入信息有误');</script>";
        }
    }
//省略了部分代码
}
$rs = mysql_query("select * from tb_demo028");
?>
```

■ 秘笈心法

心法领悟 373：update 批量更新。

在本实例中，通过批量更新将指定字段的值统一更新为一个新的值。还可以通过批量更新将指定字段更新为不同的值，完成这个操作首先需要 form 表单提交多个不同的值，然后通过 while 循环读取提交的值，并且循环执行更新语句，即可实现。

实例 374 Delete 语句删除图书信息

光盘位置：光盘\MR\08\374

高级

趣味指数：★★★★

实例说明

数据库可以长期储存数据，但是如果不能及时删除一些已经过期不再使用的数据或者冗余数据信息，会影响到 MySQL 服务器的执行效率，所以删除不再使用的数据是十分必要的。本实例通过 delete 语句实现删除图书信息，运行结果如图 8.43 所示。

删除图书信息

输入删除ID 删除

ID	书名	价格	日期
1	PHP开发实战宝典	80元	2010-07-17
2	VB开发实战宝典	80元	2010-07-17
3	JAVA开发实战宝典	79元	2010-07-17
4	JAVAWEB开发实战宝典	78元	2010-07-17

图 8.43　删除图书信息

关键技术

Delete 语句用于删除表中的数据，通常有两种形式，一是可以一次删除表中多条数据；二是一次只删除一条记录，其语法格式如下：

```
DELETE [FROM]
{table_name | view_name}
[WHERE search_conditions]
```

参数说明：

table_name：指定要删除数据的数据表的名称。

view_name：用于要删除数据的视图。

search_conditions：使用搜索条件来限定要删除的数据行。

设计过程

（1）建立 in.php 文件，通过面向对象的知识编写数据库连接类。

（2）编写 index.php 文件。首先，创建 form 表单，将删除数据的 ID 提交到本页。然后，包含数据库连接类并实例化对象，判断表单中提交的数据是否符合要求，如果符合要求，则执行 delete 删除语句。最后，循环输出数据表中数据，核心代码如下：

```
<?php
include("in.php");                                              //包含数据库连接类
new mysql("localhost","root","111","db_database08");            //实例化对象
if($_POST[sub]){                                                //单击删除按钮
    if(preg_match("/\d/",$_POST[na])){                          //首先验证文本框是否为数字
        if(mysql_query("delete from tb_demo029 where id='$_POST[na]'")){  //执行删除操作
            echo "<script>alert('删除成功');</script>";          //输出提示
        }else{
            echo "<script>alert('您输入的商品 ID 不在范围内');</script>";
        }
    }else{
        echo "<script>alert('您输入的图书 ID 不是数字');</script>";
    }
}
$rs = mysql_query("select * from tb_demo029");                  //查询操作输出信息
```

```
echo "<table width='580px'><td>ID</td><td>书名</td><td>价格</td><td>日期</td></tr>";
while($rst = mysql_fetch_row($rs)){
        echo "<td>".$rst[0]."</td><td bgcolor='#FFFFFF'>".$rst[1]."</td><td>".$rst[2]."</td><td bgcolor='#FFFFFF'>".$rst[3]."</td></tr>";
}
echo "</table>";
?>
```

秘笈心法

心法领悟 374：使用 delete 语句的注意事项。

如果用户编写的 delete 语句中没有 where 子句，则所有的字段信息都将被删除。

实例 375	数据信息的批量删除 光盘位置：光盘\MR\08\375	高级 趣味指数：★★★★

实例说明

假设在数据表中有多条错误信息或者冗余数据需要用户手动删除，如果用户逐一删除，未免浪费时间，降低用户的工作效率。本实例推荐一种高效的删除方法，实现数据的批量删除。运行本实例，如图 8.44 所示，对网络聊天室中的数据进行管理，选中一条或者多条数据，单击“删除”按钮即可实现将选中的所有数据删除。

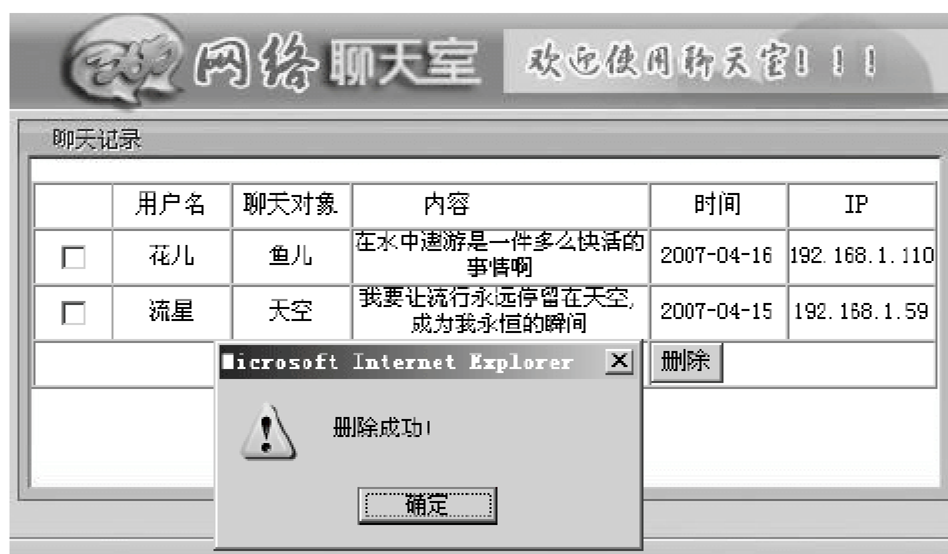


图 8.44　批量删除数据

关键技术

批量删除数据同样应用的是 delete 语句，只是在处理提交的数据时应用 while 循环语句、list()和 each()函数。list()函数的语法如下：

```
void list(…);
```

该函数用于一次性为多个变量赋值。

each()函数的语法如下：

```
array each( array array);
```

该函数用于依次返回参数 array 所指定的数组的元素下标和该下标所对应的数组元素值，并且每获取一个数组元素后，自动使数组指针向下移一位。该函数将返回一个数组，该数组的下标只有 0、key、1 和 value 这 4 个数值，其中下标为 0 和 key 的数组元素值相等，表示该函数的参数所指定数组的当前元素的下标，而下标为 1 和 value 的数组元素值相等，表示该参数所指定数组的当前元素值。

while 循环语句是 PHP 语言中常用的循环语句中的一种，该语句的基本格式如下：

```
while (expr) statement ;                              //基本格式
while (expr) : statement ... endwhile ;               //使用替代语法的格式
```

while 循环是 PHP 最简单的循环语句。只要 while 表达式的值为 TRUE 就重复执行该表达式中的语句体，如果 while 表达式的值一开始就是 FALSE，则循环语句体一次也不执行。

设计过程

（1）新建 conn.php，创建与数据库的连接，其代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111")or dir('服务器连接失败:'. mysql_error());
mysql_select_db("db_database08",$conn);
mysql_query("set names utf8");
?>
```

（2）创建显示数据库中数据的表格，读取数据库中的数据，设置批量删除数据的提交按钮，关键代码如下：

```
<?php session_start(); include("conn/conn.php");
<form name="form1" method="post" action="index_ok.php">
        <table width="486" border="1" cellpadding="0" cellspacing="0">
        <?php $query=mysql_query("select * from tb_chartroom");
        if($query==true){
        while($myrow=mysql_fetch_array($query)){
        ?>
           <tr>
              <td align="center"><span class="STYLE2">
                 <input type="checkbox" name="<?php echo $myrow[id];?>" value="<?php echo $myrow[id];?>">
              </span></td>
              <td height="25" align="center"><span class="STYLE2"><?php echo $myrow[username];?></span></td>
              <td align="center"><span class="STYLE2"><?php echo $myrow[to_username];?></span></td>
              <td align="center"><span class="STYLE2"><?php echo $myrow[text];?></span></td>
              <td align="center"><span class="STYLE2"><?php echo $myrow[data];?></span></td>
              <td align="center"><span class="STYLE2"><?php echo $myrow[ip];?></span></td>
           </tr>
        <?php }}?>
           <tr>
              <td height="25" colspan="4" align="center"> </td>
              <td colspan="2" align="left"><input type="submit" name="Submit" value="删除"></td>
              </tr>
        </table>
           </form>
```

（3）对表单提交的数据进行处理，完整的代码如下：

```
<?php session_start(); include("conn/conn.php");
if($Submit=="删除"){
      while(list($name,$value)=each($_POST)){
             $result=mysql_query("delete from tb_chartroom where id='".$name."'");
      if($result==true){
       echo "<script>alert('删除成功!'); window.location.href='index.php';</script>";
 }else{
      echo "<script>alert('失败!'); window.location.href='index.php';</script>";}
}}
?>
```

秘笈心法

心法领悟 375：制作批量删除程序的注意事项。

在制作批量删除程序时，最好建立提示页面确认是否删除，否则如果由于意外删除数据，会给用户带来不必要的损失。

实例 376	避免输出中文字符串时出现乱码 光盘位置：光盘\MR\08\376	高级 趣味指数：★★★★

实例说明

将数据库中信息显示到网页中经常会遇到这样的问题，即网页中输出的中文文字为乱码。造成该问题的主

要原因是数据库中的编码格式与页面设置的编码格式不符或者没有将数据信息统一为 GBK、GB2312 或 utf-8。本实例通过设置编码格式演示如何避免输出中文字符串时出现乱码，运行结果如图 8.45 所示。



图 8.45　错误设置编码格式输出中文

■ 关键技术

set 语句用于设置不同变量，其语法如下：

```
SET variable  assignment [, variable  assignment] ...
variable_assignment:
        user  var  name = expr
      | [GLOBAL | SESSION] system_var_name = expr
      | @@[global. | session.]system_var_name = expr
```

本实例通过 set 语句设置页面的编码格式，其代码如下：

```
set names $_POST[name];
```

■ 设计过程

编写脚本文件，命名为 index.php。创建 form 表单，将要设置的编码格式提交到本页。在本页中连接服务器和指定的数据库，根据 form 表单提交的数据设置数据库的编码格式，然后循环输出数据库中的数据，其代码如下：

```
<?php
if($_POST[sub]){                                                                        //单击“确定”按钮
        if($_POST[name] == "" || $_POST[name] == "输入编码格式"){                        //如果文本框为空或初始值
                echo "<script>alert('文本框内容不正确');</script>";
        }else{
                $conn = mysql_connect("localhost","root","111") or dir ("connect MySQL false");     //连接 MySQL 数据库
                mysql_select_db("db_database08",$conn) or die ("connect database false");         //连接数据库
                mysql_query("set names $_POST[name]");                                          //设置编码格式
                $rs = mysql_query("select * from tb_demo031");                                  //返回结果集
                echo "<table width='580px'><tr><td bgcolor='#FF0000'>ID</td><td bgcolor='#FF0000'>书名</td><td bgcolor='#FF0000'>价格</td><td
bgcolor='#FF0000'>日期</td></tr>";
                while($rst = mysql_fetch_row($rs)){                                             //循环输出
                        echo "<tr><td bgcolor='#FF0000'>".$rst[0]."</td><td bgcolor='#FF0000'>".$rst[1]."</td><td bgcolor='#FF0000'>".$rst[2].
"</td><td bgcolor='#FF0000'>".$rst[3]."</td></tr>";
                }
                echo "</table>";
        }
}
?>
```

■ 秘笈心法

心法领悟 376：SET 关键字的特点。

set 语句用于设置不同类型的变量，这些变量会影响服务器或客户端的操作。SET 关键字可以用于向用户变量或系统变量赋值。

实例 377　动态创建 MySQL 数据库　高级

光盘位置：光盘\MR\08\377　趣味指数：★★★★

实例说明

创建数据库的方法有很多，例如在 cmd 命令提示符下通过 CREATE 语句创建 MySQL 数据库，或通过 phpMyAdmin 等图形化管理工具创建数据库。本实例通过表单提交和函数实现动态创建 MySQL 数据库，运行结果如图 8.46 所示。



图 8.46　动态创建数据库

关键技术

本实例主要应用的是 SQL 语句中的 create database 语句，create database 语句的格式如下：

```
create database db_name ;
```

其中，db_name 是要创建的数据库名称，该名称必须是合法的，不能够与其他数据库重名。

本实例实现创建数据库的 SQL 语句的代码如下：

```
create database $_POST[name];
```

设计过程

编写 index.php 文件，引入 CSS 样式表，编写网页页面结构并包含图片。当单击“创建”按钮时，首先对文本框进行数据验证，然后连接 MySQL 数据库，通过 mysql_query()函数实现动态创建 MySQL 数据库，代码如下：

```
<?php
if($_GET[id] == '1'){                                                    //单击按钮
    echo "<form action='' method='post'>";                               //编写 form 表单
    echo "<input class='one'type='text' size='10' name='name' value='数据库名称'>";
    echo "   <input class='two' type='submit' name='sub' value='       '";
    echo "</form>";
    if($_POST[sub]){                                                     //单击“创建”按钮
        if($_POST[name] == "" || $_POST[name] == "数据库名称"){           //验证文本框是否为初始值或空值
            echo "<script>alert('请填写正确数据库名称');</script>";
        }else{
            if(preg_match("/\w/",$_POST[name])){                         //正则表达式限制文本框输入内容
                $conn = mysql_connect("localhost","root","111");         //连接 MySQL 数据库
                if(mysql_query("create database $_POST[name]")){         //如果创建成功
                    echo "<script>alert('创建数据库成功');</script>";       //输出提示
                }
            }else{
                echo "<script>alert('请不要使用特殊字符或中文字符');</script>";
            }
        }
    }
}
?>
```

秘笈心法

心法领悟 377：对于合法的数据库（包括 MySQL 其他操作对象）的命名有如下一些规则：

（1）名称可以由任意字母、数字、“—”或者“$”组成，可以使用上述的任意字符开头，但不能使用单独的数字，那样会造成其与数值的混淆。

（2）名称长度的限制为：数据库、表、列和索引的名称最多由 64 个字符组成，而别名最长可达 256 个字符。

（3）不能使用 MySQL 的关键字作为数据库名和表名。

实例 378	动态创建数据表 光盘位置：光盘\MR\08\378	高级 趣味指数：★★★★

实例说明

单纯地创建数据库毫无意义，因为所有的数据都是以数据表的形式存储在数据库中。本实例通过 create table 语句实现动态创建数据表，运行结果如图 8.47 所示。



图 8.47　动态创建数据表

关键技术

本实例主要应用 SQL 语句中的 create table 语句来动态创建数据表。create table 语句的语法如下：

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] table_name
```

[(create_definition,…)][table_options] [select_statement]

本实例实现动态创建数据表的 SQL 语句的代码如下：

```
create table $_POST[n](id int(4) auto_increment primary key);
```

设计过程

创建 index.php 文件，首先通过下拉列表框循环输出数据库中存储的数据，然后，选择指定的数据库，单击“确定”按钮，在弹出的文本框中添加数据表名称，并且指定编码格式，单击“确定”按钮，提交创建数据表的数据。最后，获取 form 表单提交的数据，完成数据表的创建操作，其关键代码如下：

```
<?php
if($_POST[sub1]){                                                    //单击“确定”按钮
        if($_POST[n] == "" || $_POST[n] == "数据表名称"){             //验证文本框是否为空或初始值
                echo "<script>alert('请填写正确的数据表名称');</script>";  //输出提示
        }else{
                if(preg_match("/\w/",$_POST[n])){                    //过滤中文字符和特殊字符
                        if($_POST[select1] != "111"){                //如果下拉列表框值不为 111
                                //连接 MySQL 数据库
                                $conn = mysql_connect("localhost","root","111") or die ("connect mysql false");
                                //连接数据库
                                mysql_select_db("$_SESSION[id]",$conn) or die("connect database false");
                                mysql_query("SET NAMES $_POST[select1]");     //设置编码格式
                                //如果创建数据表成功
                                if(mysql_query("create table $_POST[n](id int(4) auto_increment primary key);")){
                                //输出提示
                                        echo "<script>alert('创建数据表成功');location.href='index.php?id=2';</script>";
                                }
                        }else{
                                echo "<script>alert('请选择编码格式');</script>";
```

```
                }
            }else{
                echo "<script>alert('数据表名称不能包含特殊字符或汉字');</script>";
            }
        }
}
?>
```

秘笈心法

心法领悟 378：创建数据表的注意事项。

创建数据表时，系统要求至少包含一个字段，空数据表是不允许创建的。

实例 379　动态创建 MySQL 数据字段

光盘位置：光盘\MR\08\379

高级

趣味指数：★★★★

实例说明

数据表是由多个字段组成的，字段又分为是什么数据类型、是否设置主键、是否可以为空、选择设置编码格式等。本实例通过 SQL 语句实现动态创建数据字段，运行结果如图 8.48 所示。

图 8.48　动态创建 MySQL 数据库字段

关键技术

本实例实现动态创建数据字段的 SQL 语句的代码如下：

```
$sql1 = "create table tb_demo035(";
$sq = "`".$_POST[text1_id]."` "."int(".$_POST[text3_id].")"." "."primary key  auto_increment";
$sql = "`".$_POST["n".$b]."` ".$_POST["select_type".$b]."(".$_POST["length".$b].")"." ".$_POST["select".$b].",";
$sql2 = ");";
```

设计过程

编写脚本文件 index.php。首先，连接数据库服务器和数据库。然后，读取服务器中存储的数据库，根据选择的数据库，创建 form 表单，提交数据表名称、字段名称和编码格式。最后，获取表单中提交的数据，执行数据表中字段的添加操作，其核心代码如下：

```
if($_POST[sub1] ){                                          //单击按钮
    //拼接字符串
    $sq = "`".$_POST[text1_id]."` "."int(".$_POST[text3_id].")"." "."primary key  auto_increment";
    $_SESSION[sq] = $sq;                                    //保存于 session 中
    for($b = 0;$b < $_SESSION[id];$b++){                    //循环语句
        if($_POST["select_type".$b] == "1"){                //确定类型
            echo "<script>alert('请选择数据类型');location.href='index.php';</script>";
        }else{
            //判断字段长度值
```

```
                    if($_POST["length".$b] != "" || $_POST["length".$b] != "长度/值"){
                        if(preg_match("/\d/",$_POST["length".$b])){          //正则表达式
                            $sql = "`".$_POST["n".$b]."` ".$_POST["select_type".$b]."(".$_POST["length".$b].")"." ".$_POST["select".$b].",";
                            $_SESSION['sql'.$b] = $sql;                       //保存于 session
                        }else{
                            echo "<script>alert('不是一个数字');location.href='index.php';</script>";
                        }
                    }
                }
            }
        $sql1 = "create table tb_demo035(";                                  //定义字符串
        $sql2 = ");";                                                         //定义字符串
        $array = array();                                                     //定义数组
        for($d = 0;$d < $_SESSION[id];$d++){                                  //循环语句
            array_push($array,$_SESSION[sql.$d]);
        }
        $arr = implode($array);                                               //合并字符串
        $s = $sql1.$arr.$sq.$sql2;                                            //合成 sql 语句
        //连接 MySQL
        $conn = mysql_connect("localhost","root","111") or die("connect mysql false");
        //连接数据库。
        mysql_select_db("db_database08",$conn) or die("connect database false");
        mysql_query("set names utf8");                                        //设置编码
        if(mysql_query($s)){                                                  //执行创建
            echo "<script>alert('动态创建数据字段成功');</script>";
        }else{
            echo "<script>alert('以存在数据库')</script>";
        }
    }
?>
```

秘笈心法

心法领悟 379：动态创建字段还可以使用 SQL 中的 alter table 语句实现。

动态创建字段只是向指定的数据表中添加新的字段及字段属性值。修改表的操作语句为 alter table，主要实现对数据表结构进行操作，也可以对表中的列进行添加或删除、创建或撤销索引、更改现有列的类型或更改列或表的名称及修改表的注释和类型。该语句的格式如下：

```
alter [ignore] table table_name action_list ;
```

参数说明：

该语句允许指定多个动作，其动作间使用逗号分隔，每个 action 表示对表的一个修改。它不仅能够同时执行多个修改操作，还可以同时将所有 varchar 列更改为 char 列，从而实现将行可变的表更改为行定长的表。在表 8.2 中，列出了 alter table 语句的常用功能。

表 8.2　alter table 语句的常用功能

功　能	示 例 语 句	说　明
对表重新命名	alter table table_name rename as new_table_name ;	只需给出原表名和新表名即可
更改列的类型	alter table s_table modify id int unsigned ; alter table s_table change id int unsigned ; alter table s_table change id sid int unsigned ;	该语句将 s_table 表中的列 id 由 smallint unsigned 类型更改为 int unsigned 类型 第 3 条语句中，使用 change 既更改了 id 的类型，又重新将 id 命名为 sid
将字符串列从可变长转为定长	alter table s_table modify user char (30) ;	在字符串中，定长的列一般比变长的列处理得快。如语句中，s_table 中列 user 的类型是 varchar，将其转换为定长的 char 列，可提高查询的效率
将字符串列从定长转为可变长	alter table s_table modify user varchar (30) ;	虽然字符串使用定长的行查询效率高，但是却占用了很多空间，而使用变长的字符串列却能够节省空间，使转换的效率更高

8.6　SQL 查询语句

本节将介绍在 PHP 语言中，如何运用 SQL 语言中的查询语句查询数据库中的数据。

实例 380　**查询数值型数据**　**高级**

光盘位置：光盘\MR\08\380　趣味指数：★★★★

实例说明

在实际的项目开发过程中，经常需要在数据库中建立一些数值型字段保存信息，例如商品的数量、价格等。本实例将通过在数据库中查询库存剩余量大于 100 的商品信息，来说明查询数值型数据的方法。运行本实例，如图 8.49 所示，图中显示的为库存量大于 100 的商品。

查询数值型数据

查询数据

ID	书名	剩余	日期
1	《PHP范例大全》	101	2010-07-19
2	《JAVA范例大全》	150	2010-07-19
3	《VB范例大全》	160	2010-07-19
4	《C++范例大全》	178	2010-07-19

图 8.49　大于 100 的数据

关键技术

本实例实现查询大于 100 的数据的 SQL 语句的代码如下：

```
select * from tb_demo036 where sum > 100;
```

设计过程

编写 index.php 文件。建立数据库连接，显示所有满足库存量大于 100 的商品信息，代码如下：

```
<?php
 if($_POST[sub]){                                                              //单击按钮
        $conn = mysql_connect("localhost","root","111") or die("connect mysql false");   //连接 MySQL
        mysql_select_db("db_database08",$conn) or die("connect database false");        //连接数据库
        mysql_query("set names utf8");                                         //设置编码格式
        $rs = mysql_query("select * from tb_demo036 where sum > 100");         //返回结果集
?>
        <table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">书名</td><td
background="pic/head.JPG">剩余</td><td background="pic/head.JPG">日期</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                    //循环输出
                echo "<tr><td background='pic/head.JPG'>".$rst[0]."</td><td background='pic/head.JPG'>".$rst[1]."</td><td
background='pic/head.JPG'>".$rst[2]."</td><td background='pic/head.JPG'>".$rst[3]."</td></tr>";
        }
 }
?>
```

秘笈心法

心法领悟 380：列举相关的查询谓词。

本实例的 SQL 查询语句包含谓词“>”，查询谓词还包括“=”、“<>”、“!=”、“>”、“<”、“>=”、

“=<”、“!<”和“!>”。

实例 381　查询字符串　光盘位置：光盘\MR\08\381　高级　趣味指数：★★★★

实例说明

对字符串进行查询是项目开发过程中应用几率最高的查询，并且这种查询经常与通配符配合使用实现信息的匹配查询。运行本实例，如图 8.50 所示。

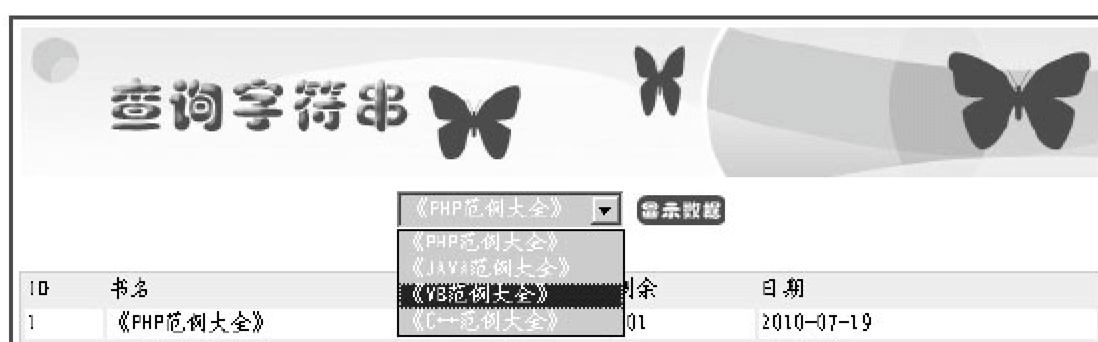


图 8.50　查询字符串

关键技术

使用 SQL 语句可以对字符串进行完全匹配查找和模糊查找，如果对字符串进行匹配查找则直接用等号作为查询条件的连接谓词，如果对字符串进行模糊查找则用 like 关键字作为连接谓词。下面将通过具体实例讲解如何实现对字符串的查找。

（1）从学生成绩表（tb_score）中查询所有名为“小刘”的学生信息，其中学生姓名字段为 sname。

```
select * from tb_score where sname='小刘'
```

（2）从学生表（tb_student）中查询所有姓“刘”的学生信息，其中学生姓名字段为 sname。

```
select * from tb_student where sname like '刘%'
```

（3）从职工表（tb_worker）中查询所有职务为“程序员”的员工信息，其中员工职务字段为 zhiwu。

```
select * from tb_worker where zhiwu like '%程序员'
```

（4）从图书信息表（tb_book）中查询所有 PHP 类相关图书，其中书名字段为 bookname。

```
select * from tb_book where bookname like '%PHP%'
```

设计过程

编写 index.php 文件，连接 MySQL 数据库，显示查询到的图书信息，代码如下：

```
<?php
if($_POST[sub]){                                                        //单击按钮
        $conn = mysql_connect("localhost","root","111") or die("connect mysql false");         //连接 MySQL
        mysql_select_db("db_database08",$conn) or die("connect database false");               //连接数据库
        mysql_query("set names utf8");                                  //设置编码格式
        //返回结果集
        $rs = mysql_query("select * from tb_demo036 where name like '%$_POST[select]%'");
?>
        <table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">书名</td><td
background="pic/head.JPG">剩余</td><td background="pic/head.JPG">日期</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                             //循环输出
                echo "<tr><td background='pic/head.JPG'>".$rst[0]."</td><td background='pic/head.JPG'>".$rst[1]."</td><td
background='pic/head.JPG'>".$rst[2]."</td><td background='pic/head.JPG'>".$rst[3]."</td></tr>";
        }
}
?>
```

秘笈心法

心法领悟 381：对关键字 like 的评价。

对字符串进行模糊查找可以使用 like 关键字。like 关键字十分重要，几乎在所有网站的站内搜索模块中都有其身影。

实例 382 查询日期型数据 高级

光盘位置：光盘\MR\08\382 趣味指数：★★★★

实例说明

对日期型数据进行查询在商业网站中得到了广泛的应用，例如查询员工的出生日期、商品的进货时间等。本实例将在图书信息表中查询图书出版日期为“2010-07-19”的图书信息，运行本实例，如图 8.51 所示。

查询日期型数据

显示数据

ID	书名	剩余	日期
1	《PHP范例大全》	101	2010-07-19
2	《JAVA范例大全》	150	2010-07-19
3	《VB范例大全》	180	2010-07-19
4	《C++范例大全》	178	2010-07-19

图 8.51　查询日期型数据

关键技术

本实例实现查询日期型数据的 SQL 语句的代码如下：

```
select * from tb_demo036 where date = '2010-07-19';
```

查询 tb_demo036 数据表中所有时间 date 等于“2010-07-19”的数据。

设计过程

建立 index.php 文件，连接 MySQL 数据库，当单击显示数据按钮时，程序会自动显示出版日期为“2010-07-19”的所有图书，代码如下：

```
<?php
if($_POST[sub]){                                                              //单击按钮
        $conn = mysql_connect("localhost","root","111") or die("connect mysql false");     //连接 MySQL
        mysql_select_db("db_database08",$conn) or die("connect database false");           //连接数据库
        mysql_query("set names utf8");                                        //设置编码格式
        $rs = mysql_query("select * from tb_demo036 where date = '2010-07-19'");           //返回结果集
?>
        <table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">书名</td><td
background="pic/head.JPG">剩余</td><td background="pic/head.JPG">日期</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                   //循环输出结果
            echo "<tr><td background='pic/head.JPG'>".$rst[0]."</td><td background='pic/head.JPG'>".$rst[1]."</td><td
background='pic/head.JPG'>".$rst[2]."</td><td background='pic/head.JPG'>".$rst[3]."</td></tr>";
        }
}
?>
```

秘笈心法

心法领悟 382：不同数据库查询日期型数据存在差异。











图 8.55　利用变量查询字符串数据

关键技术

利用变量实现数据的查询，经常需要通过表单提交、获取要查询的内容，并将该内容赋予某变量后传入 SQL 语句执行查询，并通过$_POST[]方式进行接收。

本实例中实现按图书名称进行匹配查询的代码如下：

```
$sql=mysql_query("select * from tb_demo031 where bname like '%$_POST[text] %'");
```

由于传入 SQL 语句的参数为字符型数据，所以应在传入参数的两侧加上单引号表示该数据为字符型，否则程序将发生类型不匹配的错误。当然该语句还可以写成如下形式：

```
$sql=mysql_query("select * from tb_demo031 where bname like '%'.$_POST[text].'%'",$conn);
```

这是因为 PHP 中的字符串有两种表示形式，一种是可以将字符串用单引号括起来，另一种是将字符串用双引号括起来。如果采用第二种表示形式，PHP 变量可以直接写在该字符串中，系统并不会将该变量看作是字符串的一部分，还是将其当作 PHP 的合法变量。

设计过程

编写文件 index.php，设计网页框架，连接 MySQL 数据库，根据提示当单击“查看”按钮时，将传入的字符串应用于 SQL 语句中，实现模糊查询，代码如下：

```
<?php
if($_POST[sub]){                                                                    //单击“查看”按钮
        if($_POST[text] == "" || $_POST[text] == "输入图书名称"){                   //当文本框为空或初始值
                echo "<script>alert('输入图书名');</script>";                       //输出提示
        }else{
                $conn = mysql_connect("localhost","root","111") or die("connect mysql false");      //连接 MySQL
                //连接数据库
                mysql_select_db("db_database08",$conn) or die("connect database false");
                mysql_query("set names utf8");                                      //设置编码格式
                $rs = mysql_query("select * from tb_demo031 where bname like '%$_POST[text]%'");    //返回结果集
                echo'<table width="580px"><tr><td    background="pic/head.JPG">ID</td><td    background="pic/head.JPG">书名</td><td
background="pic/head.JPG">价格</td><td    background="pic/head.JPG">日期</td></tr>';                //循环输出结果
                while($rst = mysql_fetch_array($rs)){
?>

        <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
                }
        }
}
?>
```

秘笈心法

心法领悟 386：完全匹配的方法。

本实例中使用了完全匹配符“%%”，表示查询字符串可以出现在任何位置。

实例 387 查询前 n 条记录

光盘位置：光盘\MR\08\387

高级

趣味指数：★★★★

实例说明

在图书管理系统中，经常需要查询前 n 本书的详细信息。运行本实例，如图 8.56 所示，首先在图中的文本框中输入要查询的记录个数，单击“查看”按钮即可按要求查找到满足条件的图书信息。

查询前N条记录

输入数字 查看

ID	书名	价格	日期
1	PHP开发实战宝典	80元	2010-07-17
2	VB开发实战宝典	80元	2010-07-17

图 8.56　查询前 n 条数据

关键技术

关键字 limit 是 MySQL 数据库的扩展部分，利用它可以实现查询从指定位置开始满足一定条件的 n 条记录，如果 n 超出表的范围，则只显示从指定位置开始到表结束的所有记录。本实例通过如下代码实现查找指定的前 n 条记录：

```
$sql=mysql_query("select * from tb_student where classname like '%".$classname."%' order by userid desc limit 0,$number ",$conn);
```

设计过程

编写 index.php 文件，开启 MySQL 服务，连接 MySQL 数据库，当单击“查看”按钮时，显示前 n 条记录，代码如下：

```
<?php
 if($_POST[sub]){                                                          //单击“查看”按钮
        if($_POST[text] == "" || $_POST[text] == "输入图书名称"){              //判断文本框是否为空或初始值
                echo "<script>alert('输入图书名');</script>";                   //JavaScript 提示
        }else{
                if(preg_match("/\d/",$_POST[text])){                          //正则表达式过滤字符
                        $conn = mysql_connect("localhost","root","111") or die("connect mysql false");   //连接 MySQL
                        mysql_select_db("db_database08",$conn) or die("connect database false");      //连接数据库
                        mysql_query("set names utf8");                        //设置编码格式
                        $rs = mysql_query("select * from tb_demo031 limit $_POST[text]");    //返回结果集
                        echo'<table width="580px"><tr><td   background="pic/head.JPG">ID</td><td   background="pic/head.JPG">书名</td><td
background="pic/head.JPG">价格</td><td   background="pic/head.JPG">日期</td></tr>';
                        while($rst = mysql_fetch_array($rs)){                 //循环输出
?>

        <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
                        }
                }else{
                        echo "<script>alert('输入的不是一个数字');</script>";
                }
        }
}
?>
```

秘笈心法

心法领悟 387：limit 语句的优点。

limit 语句与其他语句，例如 order by 等语句联合使用，会使用户的 SQL 语句千变万化，从而使程序非常灵活。

实例 388　查询后 n 条记录

光盘位置：光盘\MR\08\388

高级

趣味指数：★★★★

实例说明

对后 n 条记录的查询在实际项目开发过程中也经常被用到，例如查询最后添加到数据库中的 n 条信息。运行本实例，如图 8.57 所示，显示数据库最后的若干数据。

查询后N条数据

输入数字　查看

ID	书名	价格	日期
13	PHP范例宝典	75元	2010-07-17
12	PHP范例宝典	75元	2010-07-17
11	PHP范例宝典	75元	2010-07-17
10	PHP范例宝典	78元	2010-07-17
9	VB范例宝典	78元	2010-07-17

图 8.57　查询后 n 条数据

关键技术

与查询前 n 条记录相比，查询后 n 条记录首先应对表中的所有记录进行降序排列，之后通过关键字 limit 指定要查询的记录个数。对记录升序排列可以在 SQL 语句中添加如下语句实现：

```
order by 字段名(或 order by 字段名 asc)
```

对记录降序排列可以在 SQL 语句中添加如下语句实现：

```
order by 字段名 desc
```

本实例实现查询后 n 条数据的 SQL 代码如下：

```
select * from tb_demo031 order by id desc limit $_POST[text];
```

设计过程

编写 index.php 文件，通过<table>标签建立网页架构，引入头部图片，开启 MySQL 服务并连接 MySQL 数据库，当单击“查看”按钮时，显示后几条信息，代码如下：

```
if($_POST[sub]){                                                                //单击“查看”按钮
        if($_POST[text] == "" || $_POST[text] == "输入图书名称"){                //判断文本框
                echo "<script>alert('输入图书名');</script>";                      //JavaScript 提示
        }else{
                if(preg_match("/\d/",$_POST[text])){                            //正则表达式验证
                        $conn = mysql_connect("localhost","root","111") or die("connect mysql false");          //连接 MySQL
                        mysql_select_db("db_database08",$conn) or die("connect database false");               //连接数据库
                        mysql_query("set names utf8");                          //设置编码
                        $rss = mysql_query("select * from tb_demo031 order by id desc limit $_POST[text]");   //返回结果集
                        echo'<table width="580px"><tr><td  background="pic/head.JPG">ID</td><td  background="pic/head.JPG">书名</td><td
background="pic/head.JPG">价格</td><td  background="pic/head.JPG">日期</td></tr>';
                        while($rst = mysql_fetch_array($rss)){                  //循环输出
?>

        <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
```

```
<?php
                        }
                }else{
                        echo "<script>alert('输入的不是一个数字');</script>";
                }
        }
}
?>
```

秘笈心法

心法领悟 388：对 SQL 语句的评价。

灵活运用 SQL 语句，会让用户觉得很多看起来复杂的程序其实很简单。

实例 389 查询从指定位置的 n 条记录

光盘位置：光盘\MR\08\389

高级

趣味指数：★★★★

实例说明

PHP 中实现分页显示的原理就是从指定位置开始显示 n 条记录，下面通过本实例讲解如何实现这种显示方式。运行本实例，如图 8.58 所示。

查询从指定位置的前N条记录

输入开始ID　输入查询条数　查看

ID	书名	价格	日期
2	VB开发实战宝典	80元	2010-07-17
3	JAVA开发实战宝典	79元	2010-07-17
4	JAVAWEB开发实战宝典	78元	2010-07-17

图 8.58　查询从指定位置的 n 条记录

关键技术

实现从指定位置开始查询满足条件的 n 条记录，主要应用 MySQL 的扩展关键字 limit，该关键字的使用格式如下：

```
select 要查询的字段 from 表名 where 查询的条件 limit 满足条件的起始位置,记录的个数
```

关键字 limit 后有两个参数，第一个参数用于指定要满足条件记录的起始位置；第二个参数指定查询结果中满足条件的记录个数。

本实例实现查询指定位置的 n 条记录的 SQL 语句如下：

```
select * from tb_demo031    limit $_POST[text1], $_POST[text];
```

设计过程

创建 index.php 文件，编写网页框架，开启 MySQL 服务器并连接 MySQL 数据库，当单击“查看”按钮时，显示指定位置的 n 条记录，代码如下：

```
if($_POST[sub]){                                                                                    //单击按钮
        //判断文本框信息
        if($_POST[text] == "" || $_POST[text] == "输入图书名称" || $_POST[text1] == "" || $_POST[text1] == "输入查询条数"){
                echo "<script>alert('输入图书名');</script>";                                          //JavaScript 提示
        }else{
                if(preg_match("/\d/",$_POST[text])){                                                 //正则表达式验证
                        if(preg_match("/\d/",$_POST[text1])){                                        //正则表达式验证
                                $conn = mysql_connect("localhost","root","111") or die("connect mysql false");     //连接 MySQL
                                mysql_select_db("db_database08",$conn) or die("connect database false");           //连接 MySQL 服务器
```





关键技术

在 SQL 语句中，对指定范围内的数据进行查找可以通过以下两种方式实现，第一种方式，可以通过关键字 between...and...实现，语法格式如下：

```
select 要查找的字段 from 表名 where 字段名 between 初始值 and 终止值
```

第二种方式，可以通过比较运算符实现，语法格式如下：

```
select 要查找的字段 from 表名 where 字段名>初始值 and 字段名<终止值
```

设计过程

创建 index.php 文件，启动 MySQL 服务并连接 MySQL 数据库。当单击“查看”按钮时，显示指定时间段的数据，代码如下：

```
<?php
if($_POST[sub]){                                                     //单击“查看”按钮
                                                                     //判断文本框信息
    if($_POST[text1] == "" || $_POST[text1] == "输入开始日期" || $_POST[text] == "" || $_POST[text] == "输入结束日期"){
        echo "<script>alert('请输入日期');</script>";                 //JavaScript 提示
    }else{
                                                                     //验证日期格式的正则表达式

if(preg_match("/([0-9]{3}[1-9][0-9]{2}[1-9][0-9]{1}[1-9][0-9]{2}|[1-9][0-9]{3})-(((0[13578]|1[0-2])-(0[1-9]|[12][0-9]|3[01]))|((0[469]|11)-(0[1-9]|[12][0-9]|30))|(02-(0[1-9]|[1][0-9]|2[0-8])))/",$_POST[text])){
            $conn = mysql_connect("localhost","root","111") or die("connect mysql false");   //连接 MySQL
            mysql_select_db("db_database08",$conn) or die("connect database false");        //连接 MySQL 数据库
            mysql_query("set names utf8");                                                   //设置编码格式
            $rss = mysql_query("select * from tb_demo047 where date between '$_POST[text1]' and '$_POST[text]' order by date");
                                                                                             //返回结果集
            echo'<table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">书名</td><td background="pic/head.JPG">价格</td><td background="pic/head.JPG">日期</td></tr>';
            while($rst = mysql_fetch_array($rss)){                                           //循环输出结果
?>

    <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
            }
        }else{
            echo "<script>alert('日期格式错误！正确格式为：xxxx-xx-xx');</script>";           //JavaScript 提示信息
        }
    }
}
?>
```

秘笈心法

心法领悟 391：正则表达式的限制。

本实例所使用的正则表达式只是对平年，即一年 365 天的正则表达式验证。

实例 392　按月查询统计数据　　高级

光盘位置：光盘\MR\08\392　　趣味指数：★★★★

实例说明

在一些图书管理系统中，经常需要按年、月、日查询图书的出版信息。运行本实例，如图 8.61 所示，首先在图中文本框中输入要查询的月份，然后单击“查找”按钮，即可将该月份出版的图书信息查找出来。

按月查询统计数据

输入月份　查看 月份格式：xx

ID	书名	价格	日期
7	C++开发实战宝典	78元	2010-07-11
8	PHP范例宝典	78元	2010-07-17
9	VB范例宝典	78元	2010-07-17
10	PHP范例宝典	78元	2010-07-17

图 8.61　按月查询统计数据

关键技术

本实例实现按月查询统计数据的 SQL 语句如下：

```
select * from tb_demo047 where month(date) = '$_POST[text1]' order by date;
```

设计过程

创建脚本文件 index.php。通过<table>标签建立网页框架，引入网页头部图片文件，开启 MySQL 服务并连接 MySQL 数据库，当单击“查看”按钮时，显示查询信息，代码如下：

```
<?php
 if($_POST[sub]){                                                                    //单击“查看”按钮
        if($_POST[text1] == "" || $_POST[text1] == "输入月份"){                      //判断文本框信息
              echo "<script>alert('请输入月份');</script>";                          //JavaScript 提示
        }else{
              if(preg_match("/[^A-Za-z_]/",$_POST[text1])){                          //正则表达式验证文本框信息
                    $conn = mysql_connect("localhost","root","111") or die("connect mysql false");          //连接 MySQL
                    mysql_select_db("db_database08",$conn) or die("connect database false");              //连接 MySQL 数据库
                    mysql_query("set names utf8");                                   //设置编码集
                    $rss = mysql_query("select * from tb_demo047 where month(date) = '$_POST[text1]' order by date");
                    echo'<table width="580px"><tr><td    background="pic/head.JPG">ID</td><td    background="pic/head.JPG">书名</td><td
background="pic/head.JPG">价格</td><td    background="pic/head.JPG">日期</td></tr>';
                    while($rst = mysql_fetch_array($rss)){                           //循环输出结果
?>

        <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
                    }
              }else{
                    echo "<script>alert('月份格式错误！正确格式为：xx');</script>";
              }
        }
}
?>
```

秘笈心法

心法领悟 392：与年、月、日相关的查询函数。

SQL 语言中提供了如下函数，利用这些函数可以很方便地实现按年、月、日进行查询。

year(data)：该函数用于返回日期表达式 data 中的公元年份所对应数值。

month(data)：该函数用于返回日期表达式 data 中的月份所对应的数值。

day(data)：该函数用于返回日期表达式 data 中的日期所对应的数值。



秘笈心法

心法领悟 393：另外一种实现方式。

实现本实例的 SQL 语句代码还可以使用另外一种方式表达，如下所示：

```
select * from tb_demo047 where price !< $_POST[text1] order by price;
```

实例 394 查询结果不显示重复记录

光盘位置：光盘\MR\08\394

高级

趣味指数：★★★★

实例说明

在某些管理系统中，有重复记录是不可避免的，不在前台显示重复的记录可以通过 SQL 语言的关键字 distinct 实现。运行本实例，如图 8.63 所示，图书信息中价格相同的信息将被删除。



图 8.63 去掉重复项输出

关键技术

SQL 语句中，查询结果中不显示重复记录可以通过关键字 distinct 实现，该关键字使用的格式如下：

```
select distinct 字段名 from 表名 where 查询条件
```

distinct 关键字可以在查询结果中去除重复行，与其对应的还有关键字 all。在 SQL 语句中加入关键字 all，表示显示所有的记录，如果 SQL 语句中这两个关键字都不加，则查询结果将显示所有的记录。

设计过程

创建脚本文件 index.php，开启 MySQL 服务器并连接 MySQL 数据库，单击按钮显示信息，核心代码如下：

```
<?php
        if($_POST[submit]){                                                     //单击按钮
        if($_POST[text] != "" || $_POST[text] != "输入图书价格"){               //判断文本框信息
                if(preg_match("/[^a-zA-Z_]/",$_POST[text])){                    //正则表达式验证信息
                                                                                //返回结果集
                $rs = mysql_query("select distinct * from tb_demo031 where price = '$_POST[text]元'");
?>
<table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">书名</td><td background="pic/head.JPG">价格
</td><td background="pic/head.JPG">日期</td></tr>
<?php
                        while($rst = mysql_fetch_row($rs)){                     //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>

<?php
                        }
                }else{
                        echo "<script>alert('输入不是一个数值');</script>";     //JavaScript 提示
                }
        }else{
                echo "<script>alert('请输入信息');</script>";                   //JavaScript 提示
```

```
        }
}
?>
```

秘笈心法

心法领悟 394：使用关键字 distinct 的注意事项。

SQL 语句中的关键字 distinct 必须与 where 子句联合使用，否则输出的信息不会有变化。

实例 395　NOT 与谓词进行组合条件的查询

光盘位置：光盘\MR\08\395

高级

趣味指数：★★★★

实例说明

在开发一些网站时，有时需要查找不满足一定条件的所有记录，比如在人事系统中查找所有不姓刘的员工信息。运行本实例，如图 8.64 所示，首先在图中的文本框中输入不想显示的图书名称，然后单击“查看”按钮，即可将所有不为该名称的图书显示出来。

NOT与谓词进行组合条件的查询

输入图书名称　查看

ID	书名	价格	日期
2	VB开发实战宝典	80元	2010-07-17
3	JAVA开发实战宝典	79元	2010-07-17
4	JAVAWEB开发实战宝典	78元	2010-07-17
5	.NET开发实战宝典	78元	2010-07-17
6	C#开发实战宝典	78元	2010-07-17
7	C++开发实战宝典	78元	2010-07-17
9	VB范例宝典	78元	2010-07-17

图 8.64　not 与谓词进行组合条件查询

关键技术

not 与谓词进行组合所形成的表达式分别是 not between...and...、is not null 和 not in，这 3 种组合的详细说明如下：

（1）not between...and...：

```
where 字段名 not between 查询范围开始值 and 查询范围的结束值
```

该式查询所有不在该范围内的记录，这种限定条件可以改写成如下格式：

```
where 字段名<查询范围的开始值 and 字段名>查询范围的结束值
```

（2）is not null

该式根据所使用的关键字指定对空值或非空值的搜索，如果有任何操作数是 NULL，则表达式取数为 NULL。

（3）not in

该式根据使用的关键字是包含在列表内还是排除在列表外，指定表达式的搜索，搜索表达式可以是常量或列名，而列名可以是一组常量，但更多情况下是子查询。

设计过程

编写 index.php 文件。在文本框中输入内容后，单击“查看”按钮输出显示信息，代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die("connect mysql false");          //连接 MySQL
mysql_select_db("db_database08",$conn) or die ("connect database false");              //连接数据库
mysql_query("set names utf8");                                                          //设置编码格式
```







序排列可以在 SQL 语句的 order by 子句中指定关键字 desc。实现多条件查询也是通过 order by 字句实现的，只是多个限定条件之间应用逗号进行分隔，例如：

```
select 字段名 from 表名 where 查询条件 order by 字段 1 asc, 字段 2 desc……
```

设计过程

创建脚本文件 index.php。编写网页框架，开启 MySQL 服务并连接 MySQL 数据库，当单击“排序”按钮时，显示对数据进行多条件排序，代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die("connect mysql false");          //连接 MySQL
mysql_select_db("db_database08",$conn) or die ("connect database false");               //连接数据库
mysql_query("set names utf8");                                                           //设置编码格式
if($_POST[sub]){                                                                         //单击按钮
        $rs = mysql_query("select * from tb_demo054 order by mount desc,price asc");     //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">商品</td><td background="pic/head.JPG">数量
</td><td background="pic/head.JPG">价格</td><td background="pic/head.JPG">日期</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                              //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td><td
background="pic/head.JPG"><?php echo $rst[4];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 398：使用多条件排序的注意事项。

对查询信息进行多条件排序是为了共同限制记录的排序输出，一般情况下，由于不是单一条件限制，所以在输出效果上有一些差别。

实例 399　对统计结果进行排序

光盘位置：光盘\MR\08\399

高级

趣味指数：★★★★

实例说明

在商品信息管理系统中，经常需要对商品的价格、数量进行求和，然后根据所求得的值对商品进行排序，从而实现商品的排序输出。运行本实例，分别如图 8.68 所示，将单一类型商品求和并按商品数量和商品单价排序。

对统计结果进行排序

排序

ID	商品	数量	价格	日期	总价值
3	电风扇	20件	700元	2010-07-21	14000元
2	微波炉	15件	1300元	2010-07-21	19500元
1	电视	10件	2500元	2010-07-21	25000元
4	全自动洗衣机	10件	3500元	2010-07-21	35000元
6	笔记本电脑	8件	3000元	2010-07-21	24000元
5	空调	3件	5000元	2010-07-21	15000元

图 8.68　对统计结果进行排序

关键技术

SQL 语言中，对某字段所有数据求和可以通过函数 sum()实现，该函数的使用格式如下：

```
sum([all]字段名) 或 sum([distinct]字段名)
```

sum 中的参数如果为 all 加字段名，则表示对该字段中的所有记录进行求和，包括重复记录；如果 sum 中的参数为 distinct 加字段名，则表示对该字段的所有不重复记录进行求和。在 select 语句中使用 sum()函数的格式如下：

```
select sum(字段名) as 新字段名 from 表名 where 查询条件
```

本实例实现对统计结果排序的 SQL 语句如下：

```
select * from tb_demo054 order by mount desc,price asc;
```

设计过程

创建脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。当单击“排序”按钮时，显示运行结果，如图 8.68 所示，其代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die("connect mysql false");          //连接 MySQL
mysql_select_db("db_database08",$conn) or die ("connect database false");              //连接 MySQL 数据库
mysql_query("set names utf8");                                                          //设置编码格式
if($_POST[sub]){                                                                        //单击按钮
	$rs = mysql_query("select * from tb_demo054 order by mount desc,price asc");        //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">商品</td><td background="pic/head.JPG">数量
</td><td background="pic/head.JPG">价格</td><td background="pic/head.JPG">日期</td><td background="pic/head.JPG">总价值</td></tr>
<?php
	while($rst = mysql_fetch_row($rs)){                                                 //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?>件</td><td background="pic/head.JPG"><?php echo $rst[3];?>元</td><td
background="pic/head.JPG"><?php echo $rst[4];?><td background="pic/head.JPG"><?php echo $rst[2]*$rst[3];?>元</td></td></tr>
<?php
	}
}
?>
```

秘笈心法

心法领悟 399：求和函数。

在对数据库中的数据进行计算时会经常用到求和函数，使用该函数可以免去手动计算某些数据的麻烦，同时也提高了计算的效率和准确性。



实例 400	单列数据分组统计 光盘位置：光盘\MR\08\400	高级 趣味指数：★★★★

实例说明

在动态网站开发过程中，经常要求把某一字段的内容按照一定的规定进行分组，然后将分组后的结果按某一字段进行统计求和，最终得到查询结果。运行本实例，如图 8.69 所示，单击图中的“排序”按钮，即可实现先对表中的所有商品的单价与数量进行求和，然后按商品 ID 进行统计。

图 8.69　单列数据分组统计

关键技术

实现单列数据的分组统计，首先需要根据某字段进行分组，然后对分组结果进行统计，本实例中实现单列数据分组统计的代码如下：

```
select id,name,sum(price)as title ,date from tb_demo054 group by id order by title desc;
```

设计过程

创建脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。当单击“排序”按钮时，显示单列数据分组统计结果，如图 8.69 所示，代码如下：

```
<?php
 $conn = mysql_connect("localhost","root","111") or die("connect mysql false");                    //连接 MySQL
 mysql_select_db("db_database08",$conn) or die ("connect database false");                        //连接 MySQL 数据库
 mysql_query("set names utf8");                                                                    //设置编码集
 if($_POST[sub]){                                                                                  //单击按钮
        $rs = mysql_query("select id,name,sum(price)as title ,date from tb_demo054 group by id order by title desc");         //返回结果集
?>
 <table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">商品</td><td background="pic/head.JPG">总额
</td><td background="pic/head.JPG">日期</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                                        //循环输出结果
?>
 <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?>元</td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
        }

 }
?>
```

秘笈心法

心法领悟 400：使用分组语句的注意事项。

当分组语句 group by 和排序语句 order by 同时出现在 SQL 语句中时，要将分组语句书写在排序语句的前面，否则会出现如图 8.70 所示的错误。

图 8.70　将 order by 书写在 group by 前面

实例 401	多列数据分组统计 光盘位置：光盘\MR\08\401	高级 趣味指数：★★★★

实例说明

对商品销售结果进行统计时，经常需要按商品销售表中的多列进行分组统计。运行本实例，如图 8.71 所示，

单击图中的“排序”按钮，即可将商品销售表中的所有同名商品按销售数量和单价求和并显示统计结果。

多列数据分组统计

排序

ID	商品	总额	日期
4	全自动洗衣机	35000元	2010-07-21
1	电视	25000元	2010-07-21
6	笔记本电脑	24000元	2010-07-21
2	微波炉	19500元	2010-07-21
5	空调	15000元	2010-07-21
3	电风扇	14000元	2010-07-21

图 8.71　多列数据分组统计

关键技术

多列数据的分组统计与单列数据的分组统计实现方式类似，只是按多个字段进行分组统计时，统计字段间需用逗号进行分隔，例如：

```
select*, sum(字段 1) as 新字段 1 from  表名  group by  字段名  where  查询条件
```

本实例中实现多列数据分组统计的代码如下：

```
select *,sum(pirce*mount) as title from tb_demo054 group by id order by title desc;
```

设计过程

编写 index.php 文件，开启 MySQL 服务并连接 MySQL 数据库。当单击“排序”按钮时，显示如图 8.71 所示的内容，代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die("connect mysql false");                        //连接 MySQL
mysql_select_db("db_database08",$conn) or die ("connect database false");                             //连接 MySQL 数据库
mysql_query("set names utf8");                                                                         //设置编码格式
if($_POST[sub]){                                                                                        //单击按钮
        $rs = mysql_query("select id,name,sum(price*mount)as title ,date from tb_demo054 group by id order by title desc ");        //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">商品</td><td background="pic/head.JPG">总额
</td><td background="pic/head.JPG">日期</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                                             //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?>元</td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 401：使用 group by 语句的注意事项。

group by 语句后面一般为不是聚合函数的数列，即不是要分组的列。

实例 402	多表分组统计 光盘位置：光盘\MR\08\402	高级 趣味指数：★★★★

实例说明

开发一个大型的服务类网站，经常需要建立多张表，并且表与表之间通过某种关系建立关联。运行本实例，



使用聚集函数对学生成绩进行汇总

对学生成绩汇总

ID	学生姓名	语文成绩	数学成绩	英语成绩	总成绩
2	小潘	96	95	94	285
1	小杨	97	96	80	273
3	小刘	90	90	90	270
4	小黄	60	60	60	180

图 8.73　对学生成绩汇总

关键技术

SQL 语句中，对某字段的所有记录进行求和可以通过函数 sum()实现，该函数的使用方法如下：

```
sum([all | distinct] expression)
```

参数说明：

all：表示对指定字段的所有值进行聚集函数运算，all 为默认值，如果不加参数 all 或 distinct，则表示对指定字段的所有记录进行聚集运算。

distinct：表示对指定字段的所有非重复记录进行求和。

expression：是精确数字或近似数字数据类型分类（bit 数据类型除外）的表达式。

sum()函数的返回类型如表 8.3 所示。

表 8.3　sum()函数的返回结果

表达式结果	返 回 类 型
整数分类	int
decimal 分类	decimal(38,s) 除以 decimal(10,0)
money 和 smallmoney 分类	money
float 和 real 分类	float

设计过程

编写脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库，当单击“对学生成绩汇总”按钮时，显示如图 8.73 所示的信息，核心代码如下：

```
<?php
 $conn = mysql_connect("localhost","root","111") or die ("connect mysql false");          //连接 MySQL
mysql_select_db("db_database08",$conn) or die ("connect database false");                 //连接 MySQL 数据库
 mysql_query("set names utf8");                                                           //设置编码格式
 if($_GET[page] == "1"){                                                                  //page 等于 1 时
        $rss = mysql_query("select *,sum(chinese+math+english) as a from tb_demo059 group by name order by a desc ");   //返回结果集
?>
 <table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">学生姓名</td><td background="pic/head.JPG">语文成绩</td><td background="pic/head.JPG">数学成绩</td><td background="pic/head.JPG">英语成绩</td><td background="pic/head.JPG">总成绩</td></tr>
<?php
                while($rst = mysql_fetch_row($rss)){                                      //循环输出结果
?>
                <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td><td background="pic/head.JPG"><?php echo $rst[4];?></td><td background="pic/head.JPG"><?php echo $rst[5]?></td></tr>
<?php
        }
 }
?>
```

秘笈心法

心法领悟 403：对求和函数 sum()的理解。

sum()函数主要用于对数据表字段求和，一般与分组函数联合使用，但是只有出现在多表查询之中才能真正体现出它的优势。

实例 404　使用聚集函数 avg()求学生的平均成绩

光盘位置：光盘\MR\08\404

高级

趣味指数：★★★★

实例说明

获取统计数据平均值的方法有多种，例如可以先求得所有数据的总和，然后再除以总个数，而 SQL 语言中提供了专门求平均数的函数 avg()。本实例将利用该函数求得学生成绩表中各科成绩的平均值。运行本实例，如图 8.74 所示，单击图中 “对学生成绩汇总” 按钮，即可将各科的平均成绩显示出来。

使用聚集函数AVG求学生的平均成绩

对学生成绩汇总

ID	学生姓名	语文成绩	数学成绩	英语成绩
1	小杨	97	96	80
2	小潘	96	95	94
3	小刘	90	90	90
4	小黄	60	60	60
各科平均成绩		85.7500	85.2500	85.2500

图 8.74　取得学生的各科平均成绩

关键技术

获取某字段的所有记录的平均值，可以通过函数 avg()实现，该函数的使用方法如下：

```
avg([all | distinct] expression)
```

本实例实现利用 avg()函数实现显示平均成绩的 SQL 语句如下：

```
select avg(chinese),avg(math),avg(english) from tb_demo059;
```

设计过程

创建脚本文件 index.php。开启 MySQL 服务并连接 MySQL 数据库，当单击“对学生成绩汇总”按钮时，显示如图 8.74 所示的内容。实现此功能的核心代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die ("connect mysql false");              //连接 MySQL
mysql_select_db("db_database08",$conn) or die ("connect database false");                    //连接 MySQL 数据库
mysql_query("set names utf8");                                                               //设置编码集
if($_GET[page] == "1"){                                                                      //如果 page 等于 1.
        $rss = mysql_query("select * from tb_demo059");                                      //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">学生姓名</td><td background="pic/head.JPG">
语文成绩</td><td background="pic/head.JPG">数学成绩</td><td background="pic/head.JPG">英语成绩</td></tr>
<?php
                while($rst = mysql_fetch_row($rss)){                                         //循环输出结果
?>
                <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td><td
background="pic/head.JPG"><?php echo $rst[4];?></td></tr>
<?php
        }
        $rts = mysql_query("select avg(chinese),avg(math),avg(english) from tb_demo059");    //返回结果集
        $rsth = mysql_fetch_row($rts);
?>
<tr><td background="pic/head.JPG">各科平均成绩</td><td background="pic/head.JPG"></td><td background="pic/head.JPG"><?php echo
$rsth[0];?> </td><td background="pic/head.JPG"><?php echo $rsth[1];?></td><td background="pic/head.JPG"><?php echo $rsth[1];?></td></tr>
<?php
```

```
}
?>
```

秘笈心法

心法领悟 404：格式化函数 number_formate()的使用。

本实例并没有对平均成绩进行格式化，如果用户想对成绩进行格式化，可以使用函数 number_format()实现，其格式为：

```
string number_format ( float number [, int decimals])
```

参数说明：

number：表示要进行格式化输出的数字。

decimals：表示要保留的位数。

实例 405 使用聚集函数 min()求利润最少的商品

光盘位置：光盘\MR\08\405

高级

趣味指数：★★★★

实例说明

在销售管理系统中，商家会根据不同商品的销售利润来确定该商品的进货数量。运行本实例，如图 8.75 所示，单击图中的“获取利润最小的商品信息”按钮，即可将销售利润最小的商品信息显示在图中最后一行。

使用聚集函数MIN求利润最少的商品

获取利润最小的商品信息

ID	商品名称	进货价格/元	销售价格/元	进货数量/件	进货日期
1	电视	1500	1600	10	2010-07-22
2	冰箱	3100	3300	5	2010-07-22
3	空调	4500	4900	1	2010-07-22
4	笔记本电脑	3000	3500	8	2010-07-22
利润最小的商品	电视	100			

图 8.75　利润最小的商品输出

关键技术

SQL 语言中，获取某字段的最小值可以通过函数 min()实现，该函数的使用方法如下：

```
min([all | distinct] expression)
```

设计过程

创建脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。当单击“获取利润最小的商品信息”按钮时，显示如图 8.75 所示的内容，核心代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die ("connect mysql false");            //连接 MySQL
mysql_select_db("db_database08",$conn) or die ("connect database false");                 //连接 MySQL 数据库
mysql_query("set names utf8");                                                              //设置编码格式
$rss = mysql_query("select * from tb_demo061");                                             //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">ID</td><td background="pic/head.JPG">商品名称</td><td background="pic/head.JPG">
进货价格/元</td><td background="pic/head.JPG">销售价格/元</td><td background="pic/head.JPG">进货数量/件</td><td
background="pic/head.JPG">进货日期</td></tr>
<?php
        while($rst = mysql_fetch_row($rss)){                                                //循环输出结果
?>
        <tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td><td
background="pic/head.JPG"><?php echo $rst[4];?></td><td background="pic/head.JPG"><?php echo $rst[5];?></td></tr>
```

```
<?php
            }
if($_GET[page] == "1"){                                                          //当 page 等于 1 时
        $sql = "select min(outshop-inshop),name from tb_demo061 ";               //sql 语句
        $rs = mysql_query($sql);                                                 //返回结果集
        $rst = mysql_fetch_row($rs);                                             //输出结果
?>
<tr><td background="pic/head.JPG">利润最小的商品</td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"></td><td background="pic/head.JPG"></td><td
background="pic/head.JPG"></td></tr></table></td></tr></table></td></tr></table>
<?php
}
?>
```

秘笈心法

心法领悟 405：聚合函数 min()的主要应用。

聚集函数 min()被广泛应用于电子商务系统中，主要用于商家在进货之前计算自己商品的保底利润，也就是最小利润，以防止因市场价格变动而造成损失。



实例 406	使用聚集函数 max()求销售利润最高的商品 光盘位置：光盘\MR\08\406	高级 趣味指数：★★★★

实例说明

在商品销售管理系统中，商品的销售利润统计模块是必不可少的。运行本实例，如图 8.76 所示，在图中文本框中输入某月份，单击“查询”按钮，即可将 2005 年销售额最大的商品信息显示出来。

使用聚集函数MAX求利润最高的商品

请输入月份：12　查询

商品ID	名称	销售额	商店名称	销售日期	销售员
1	钻戒	5570	祥云珠宝行	2005-12-08	张林
2	18k吊坠	320402	祥云珠宝行	2005-12-11	小李
366	PT900耳钉	215456	东方广场	2005-12-05	孙丽
367	18k钻石爱链	3000	东方广场	2005-12-07	刘芳
2	18k吊坠	320402	祥云珠宝行	2005-12-11	小李

图 8.76　获取月利润最大的商品信息

关键技术

实现本实例，主要应用到 SQL 语言中的 max()函数，该函数的语法格式如下：

```
max([all | distinct] expression)
```

参数说明：

all：表示对指定字段的所有值进行聚集函数运算，all 为默认值，如果不加参数 all 或 distinct，则表示对指定字段的所有记录进行聚集运算。

distinct：表示对指定字段的所有非重复记录求最大值。

expression：是精确数字或近似数字数据类型（bit 数据类型除外）分类的表达式。

设计过程

（1）建立 conn.php 文件，实现与数据库的连接，代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");                 //连接 MySQL 数据库
```



通过 count()函数求出表中所有记录的总和。本实例将讲解 SQL 语言中 count()函数的应用。运行本实例，如图 8.77 所示，首先在图中的文本框中输入某数值，然后单击“查询”按钮，即可将所有利润大于该数值的数据求出。

取得大于某值的数据

1600 查询

ID	商品名称	进货价格/元	销售价格/元	进货数量/件	进货日期
1	电视	1500	1600	10	2010-07-22
2	冰箱	3100	3300	5	2010-07-22
3	空调	4500	4900	1	2010-07-22
4	笔记本电脑	3000	3500	8	2010-07-22

共查找到3条

图 8.77　取得大于某值的数据

关键技术

统计表中满足一定条件的记录个数，可以通过 SQL 语言的 count()函数实现，该函数的使用格式如下：

```
max([all | distinct] expression)
```

该函数参数的详细说明请详见实例 403，这里不再赘述。

本实例实现取得大于某值的数据的 SQL 语句如下：

```
select count(*) as a from tb_demo061 where $_POST[text] < outshop;
```

设计过程

创建脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库，当在文本框中输入内容时，单击“查询”按钮，显示如图 8.77 所示的内容，核心代码如下：

```
<?php
        if($_POST[sub]){                                                                    //单击按钮
            if($_POST[text] != "" || $_POST[text] != "输入查询值"){                          //判断文本框信息
                if(preg_match("/\d/",$_POST[text])){                                        //正则表达式验证信息
                    $rs = mysql_query("select count(*) as a from tb_demo061 where $_POST[text] < outshop");   //返回结果集
                    $rst = mysql_fetch_row($rs);                                            //输出结果
?>
<table><tr><td width="580px" align="center">共查找到<?php echo $rst[0];?>条</td></tr></table>
<?php
                }else{
                    echo "<script>alert('请正确输入');</script>";                            //JavaScript 提示
                }
            }else{
                echo "<script>alert('请在文本框中输入内容');</script>";                      //JavaScript 提示
            }
        }
?>
```

秘笈心法

心法领悟 407：分析 count()聚集函数与 mysql_num_rows()函数。

count()聚集函数与 PHP 操作 MySQL 函数 mysql_num_rows()在处理分页显示时，作用完全相同，但是后者功能较单一，不够灵活。

实例 408	聚集函数 first()和 last()的使用 光盘位置：光盘\MR\08\408	高级 趣味指数：★★★★

实例说明

SQL 语言中，获取表中的第一条记录或最后一条记录可以通过函数 first()和 last()实现，由于 Access 数据库

对这两个函数的支持较为完善，本实例将采用 Access 作为后台数据库。运行本实例，如图 8.78 所示，分别单击图中的“获取第一条记录”和“获取最后一条记录”按钮即可获得表中的第一条或最后一条记录。

聚集函数First和Last的使用

获取第一条记录 获取最后一条记录

学号	姓名	班级	科目	成绩
0312106	小潘	三年一班	物理	12

图 8.78 聚集函数 first()和 last()的使用

关键技术

SQL 语言中，获取表中的第一条记录或最后一条记录可以通过函数 first()和 last()实现，这两个函数的使用语法如下：

```
first(expression)
last(expression)
```

expression 代表一个字符串表达式，它包含要使用的数据字段，或者是使用该数据字段参与执行的表达式，expression 中的操作可以包含表的字段、常量等。

设计过程

（1）建立 conn.php 文件，实现与数据库的连接，代码如下：

```
<?php
$conn=new com("adodb.connection");                                                    //实例化 adodb.connection
$connstr="provider=microsoft.jet.oledb.4.0;data source=".realpath("./data/db_database07_186.mdb");    //调用 db_database07_196.
$conn->open($connstr);
?>
```

（2）判断用户提交的名称，从而确定显示第一条记录还是最后一条记录，代码如下：

```
     <?php
  function convert($text){                                                              //自定义转码函数
      return iconv('gbk', 'utf-8', $text);
  }
  if($_POST[submit1]!="" || $_POST[submit2]!="") {
      include("conn.php");                                                              //包含连接数据库文件
       $rs=new com("adodb.recordset");                                                  //实例化对象
       if($_POST[submit1]!="")                          {
       $rs->open("select first(sno) as sno1 ,first(sname) as sname1, first(sclass) as sclass1,first(ssubject) as ssubject1,first(sscore) as sscore1 from
score",$conn,3,1);
       }elseif($_POST[submit2]!=""){
       $rs->open("select last(sno) as sno1 ,last(sname) as sname1, last(sclass) as sclass1,last(ssubject) as ssubject1,last(sscore) as sscore1 from
score",$conn,3,1);
       }
       if($rs->eof|| $rs->bof){
       ?>
     <tr>
       <td height="20" colspan="5" bgcolor="#FFFFFF"><div align="center">没有查找到该学生成绩！</div></td>
     </tr>
     <?php
       } else{

  ?>
     <?php
  while(!$rs->eof){
  ?>
     <tr>
       <td width="170" height="20" bgcolor="#FFFFFF"><div align="center">
         <?php echo convert($rs->fields("sno1")->value)?>
       </div></td>
       <td width="170" bgcolor="#FFFFFF"><div align="center">
         <?php echo convert($rs->fields("sname1")->value)?>
       </div></td>
       <td width="143" bgcolor="#FFFFFF"><div align="center">
         <?php echo convert($rs->fields("sclass1")->value)?>
```

```
            </div></td>
            <td width="130" bgcolor="#FFFFFF"><div align="center">
                <?php echo convert($rs->fields("ssubject1")->value)?>
            </div></td>
            <td width="131" bgcolor="#FFFFFF"><div align="center">
                <?php echo convert($rs->fields("sscore1")->value)?>
            </div></td>
        </tr>
        <?php
        $rs->movenext;
            }
        }
}
?>
```

秘笈心法

心法领悟 408：Access 数据库——PHP 程序员必修课程之一。

Access 数据库并不是与 PHP 操作数据库联合使用的首选，但是由于各个数据库之间存在差异，函数支持程度也有强有弱，所以作为一个优秀的 PHP 程序员，不仅要掌握 MySQL 这一种数据库，还要对 Access 数据库有所了解。

8.7　多 表 查 询

在实际的 Web 开发过程中，经常将类别不同的信息存储在不同的表中，表之间通过特定的字段相互关联，所以在任何一个有规模的 Web 项目中，都可以看到多表查询的身影。本节将通过几个典型的实例讲解 SQL 语言中多表查询的使用方法。

实例 409	使用 select 子句进行多表查询 光盘位置：光盘\MR\08\409	高级 趣味指数：★★★★

实例说明

在实际项目开发过程中，经常需要将不同的信息存储在不同的表中，表与表之间通过某字段相互关联，从而使表的指针形成一种连动关系进而可以通过 SQL 语言的 select 语句实现多表查询。运行本实例，如图 8.79 所示，在图中文本框中输入要查询成绩的学生学号，单击“查找”按钮，即可将学生信息和该学生的各科成绩查找出来。

使用form子句进行多表查询

输入学生ID　查询

学生ID	学生姓名	家庭住址	入学日期	数学考试分数	英语考试分数	语文考试分数
1	小杨	吉林长春	2010-07-22	92	80	99

图 8.79　多表查询

关键技术

SQL 语言中，通过 where 子句实现多表查询，所要查找的字段名最好用“表明.字段名”表示，这样可以防止因表之间字段重名而造成无法获知该字段属于哪个表。在 where 子句中，多个表之间所形成的连动关系应按如下形式书写：

```
表 1.字段=表 2.字段  and  其他查询条件
```

综上所述，实现多表查询，应按如下形式：

```
select 字段名 from 表 1,表 2…where 表 1.字段=表 2.字段 and 其他查询条件
```

设计过程

编写 index.php 文件，当用户在文本框中输入数据并单击“查询”按钮时，显示如图 8.79 所示的内容，核心代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die ("Connect MySQL false");            //连接 MySQL
mysql_select_db("db_database08",$conn);                                                    //连接数据库
mysql_query("SET NAMES UTF8");                                                             //设置编码格式
if($_POST[sub]){                                                                           //当单击按钮时
                                                                                           //返回结果集。
	$rs = mysql_query("select a.id,a.name,a.address,a.date,b.math,b.english,b.chinese from tb_demo065_tel as b,tb_demo065 as a where a.id=b.id
and a.id='$_POST[text]'");
?>
<table width="580px"><tr><td background="pic/head.JPG">学生 ID</td><td background="pic/head.JPG">学生姓名</td><td
background="pic/head.JPG">家庭住址</td><td background="pic/head.JPG">入学日期</td><td background="pic/head.JPG">数学考试分数</td><td
background="pic/head.JPG">英语考试分数</td><td background="pic/head.JPG">语文考试分数</td></tr>
<?php
	while($rst = mysql_fetch_row($rs)){                                                    //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td><td
background="pic/head.JPG"><?php echo $rst[4];?></td><td background="pic/head.JPG"><?php echo $rst[5];?></td><td
background="pic/head.JPG"><?php echo $rst[6];?></td></tr>
<?php
	}
}
?>
```

秘笈心法

心法领悟 409：设置两表关联的注意事项。

本实例在 where 子句上，以两张表的 id 字段信息相同作为条件建立两表关联。笔者意在显示实例原理，在实际开发中不建议这样使用，最好使用主外键约束来实现。

实例 410	使用表的别名进行多表查询 光盘位置：光盘\MR\08\410	高级 趣味指数：★★★★

实例说明

为了提高 SQL 语句的可读性，经常采取为表取别名的方式对表进行查询。运行本实例，如图 8.80 所示，首先在图中文本框中输入要查询的学生成绩的学号，然后单击“查询”按钮，即可将该学生的学生成绩查找出来。

图 8.80　使用表的别名

关键技术

SQL 语言中，可以通过两种方式为表指定别名，第一种是通过关键字 as 指定，例如：

```
select 字段名 from 表名 as 表的别名 where 条件...
```

第二种是在表名后直接加表的别名从而为表指定别名，例如：

```
select 字段名 from 表名 表的别名 where 条件...
```

本实例将利用表的别名实现多表查询，代码如下：

```
select a.id,a.name,a.address,a.date,b.math,b.english,b.chinese from tb_demo065_tel as b ,tb_demo065 as a where a.id=b.id and a.id='$_POST[text]';
```

设计过程

创建网页文件 index.php。编写网页框架，开启 MySQL 服务并连接 MySQL 数据库。当在文本框中输入信息，单击“查询”按钮时，显示如图 8.80 所示的内容，核心代码如下：

```
<?php
 $conn = mysql_connect("localhost","root","111") or die ("Connect MySQL false");          //连接 MySQL
 mysql_select_db("db_database08",$conn);                                                   //连接 MySQL 数据库
 mysql_query("SET NAMES UTF8");                                                            //设置编码格式
 if($_POST[sub]){                                                                          //单击按钮
        $rs = mysql_query("select a.id,a.name,a.address,a.date,b.math,b.english,b.chinese from tb_demo065_tel as b ,tb_demo065 as a where a.id=b.id
and a.id='$_POST[text]'");                                                                 //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">学生 ID</td><td background="pic/head.JPG">学生姓名</td><td
background="pic/head.JPG">家庭住址</td><td background="pic/head.JPG">入学日期</td><td background="pic/head.JPG">数学考试分数</td><td
background="pic/head.JPG">英语考试分数</td><td background="pic/head.JPG">语文考试分数</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                                //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td><td
background="pic/head.JPG"><?php echo $rst[4];?></td><td background="pic/head.JPG"><?php echo $rst[5];?></td><td
background="pic/head.JPG"><?php echo $rst[6];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 410：使用表的别名。

在使用表的别名时应注意以下两点：

（1）别名通常是一个缩短了的表名，用于在连接中引用表中的特定列，如果连接中的多个表中有相同的名称列存在，必须使用表名或表的别名限定列名。

（2）如果定义了表的别名就不能再使用表名。

实例 411　合并多个结果集

光盘位置：光盘\MR\08\411

高级

趣味指数：★★★★

实例说明

SQL 语言中，使用关键字 union 可以将多张表的信息合并输出。如图 8.81 所示为数据库中无关联的两张员工信息表，运行本实例，单击“合并”按钮，即可将这两张表的信息合并输出，如图 8.82 所示。

uid	name	price	date
1	电视	15100	2010-07-24
2	空调	5000	2010-07-24
3	全自动洗衣机	3500	2010-07-24
4	热水器	1200	2010-07-24

id	name	pwd
1	小杨	1081
2	小潘	1026
3	小刘	1015
4	小黄	1077

图 8.81　两张无关联表

多表查询合并多个结果集

合并

ID	名称	说明
1	电视	15100
2	空调	5000
3	全自动洗衣机	3500
4	热水器	1200
1	小杨	1081
2	小潘	1026
3	小刘	1015
4	小黄	1077

图 8.82　合并多个结果集





关键技术

多表之间的嵌套查询可以通过谓词 in 实现，该谓词的语法格式如下：

```
test_expression[not] in{
  subquery
}
```

参数说明：

test_expression：指 SQL 表达式。

subquery：包含某列结果集的子查询，该列必须与 test_expression 具有相同的数据类型。

设计过程

创建脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。当单击“查询”按钮时，显示如图 8.84 所示的内容，代码如下：

```
<?php
 $conn = mysql_connect("localhost","root","111") or die ("Connect MySQL False");                    //连接 MySQL
 mysql_select_db("db_database08",$conn) or die ("Connect Database False");                          //连接 MySQL 数据库
 mysql_query("SET NAMES UTF8");                                                                     //设置编码
 if($_POST[sub]){                                                                                   //设置编码格式
        $sql = "select * from tb_demo069_people where uid in (select deptID from tb_demo069_dept where deptName = '$_POST[text]')";      //sql 语句
        $rs = mysql_query($sql);                                                                    //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">员工编号</td><td background="pic/head.JPG">部门编号</td><td
background="pic/head.JPG">员工姓名</td><td background="pic/head.JPG">性别</td></tr>
<?php
              while($rst = mysql_fetch_row($rs)){                                                   //循环输出
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
        }
 }
?>
```

秘笈心法

心法领悟 413：多表嵌套查询的原理。

对于数据表的嵌套查询，用户一定要掌握其查询原理，即无论是多少张表进行嵌套，表与表之间一定存在某种关联，通过 where 子句建立此种关联实现查询。

实例 414　嵌套查询在查询统计中的应用

光盘位置：光盘\MR\08\414

高级　趣味指数：★★★★

实例说明

嵌套查询在实际项目开发过程中应用较频繁，可以通过一条 SQL 语句实现多表之间的复杂查询。本实例将利用嵌套查询实现查询部门员工与公司部门的详细信息。运行本实例，如图 8.85 所示。

图 8.85　查询统计应用

关键技术

实现多表的嵌套查询时，可以同时使用谓词 any、some 和 all，这些谓词被称为定量比较谓词，可以和比较运算符联合使用，判断是否全部返回值都满足搜索条件。some 和 any 谓词是存在量的，只注重是否有返回值满足搜索要求，这两个谓词的含义相同，可以替换使用；all 谓词称为通用谓词，它只关心是否有谓词满足搜索要求。

any | some |all 谓词的语法形式如下：

```
scalar_expression {查询条件运算符} { some | any }(subquery)
```

参数说明：

scalar_expression：指任何有效的 SQL 表达式

subquery：指包含某列结果集的子查询。

本实例实现查询统计的 SQL 语句的代码如下：

```
select * from tb_demo069_people where uid in (select deptID from tb_demo069_dept where deptName = '$_POST[select]';
```

设计过程

创建脚本文件 index.php，编写下拉列表框。选择部门，当单击“查询”按钮时，显示如图 8.85 所示的内容，代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die ("Connect MySQL False");            //连接 MySQL
mysql_select_db("db_database08",$conn) or die ("Connect Database False");                  //连接 MySQL 数据库
mysql_query("SET NAMES UTF8");                                                              //设置编码格式
if($_POST[sub]){                                                                            //单击按钮
        if($_POST[select] == "1"){                                                          //下拉列表设置
                echo "<script>alert('请选择部门');</script>";                                //JavaScript 设置
        }else{
                //sql 语句
                $sql = "select * from tb_demo069_people where uid in (select deptID from tb_demo069_dept where deptName = '$_POST[select]')";
                $rs = mysql_query($sql);                                                    //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">员工编号</td><td background="pic/head.JPG">部门名称</td><td
background="pic/head.JPG">员工姓名</td><td background="pic/head.JPG">性别</td></tr>
<?php
                while($rst = mysql_fetch_row($rs)){                                         //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $_POST[select];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
                }
        }
}
?>
```

秘笈心法

心法领悟 414：嵌套查询的广泛应用。

利用谓词 all 查询所有员工工资在 3000 元以上的员工信息。

利用谓词 any 查询由人民邮电出版社出版的图书名称。

```
scalar_expression {查询条件运算符} { some | any }(subquery)
```

参数说明：

scalar_expression：指任何有效的 SQL 表达式

subquery：指包含某列结果集的子查询



SQL 语言中，应用子查询应遵循以下规则：

（1）由比较运算符引入的内层子查询只包含一个表达式或列名，在外层语句中的 where 子句内命名的列必须与内层子查询命名的列兼容。

（2）由不可更改的比较运算符引入的子查询（比较运算符后面不跟关键字 any 或 all）不包括 group by 或 having 子句，除非预先确定了成组或单个的值。

（3）用 exists 引入的 select 列表一般都由*组成，不必指定列名。

（4）子查询不能在内部处理其结果。

实例 416　使用子查询作表达式

光盘位置：光盘\MR\08\416

高级　趣味指数：★★★★

实例说明

本实例通过计算各科平均成绩作为子查询表达式的方式来取得学生的各科平均成绩。运行本实例，如图 8.88 所示，单击图中的“显示平均成绩”按钮，即可将该班级学生的所有学科的平均成绩显示在图中最后一行。

使用子查询作派生的表

显示平均成绩

学生ID	学生姓名	学生性别	学生年龄	语文成绩	数学成绩	英语成绩
1	小杨	男	26	94	92	99
2	小刘	男	27	99	94	93
3	小潘	男	27	98	94	93
各科平均成绩				97.0000	93.3333	95.0000

图 8.88　使用子查询作派生的表

关键技术

本实例中，求得学生各科平均成绩的代码如下：

```
select (select avg(chinese) from tb_demo071),(select avg(math) from tb_demo071),(select avg(english) from tb_demo071) from tb_demo071;
```

其中，select avg(chinese) from tb_demo071、select avg(math) from tb_demo071 和 select avg(english) from tb_demo071 为该 SQL 语句的子查询，分别用于获得该班级学生的语文、数学和英语的平均成绩。

设计过程

创建脚本文件 index.php。通过<table>标签编写网页文件，开启 MySQL 服务并连接 MySQL 数据库。当单击“显示平均成绩”按钮时，显示如图 8.88 所示的内容，核心代码如下：

```
<?php
	if($_GET[id] == "1"){                                                  //如果地址栏 id 等于 1
		//sql 语句
		$sql = "select (select avg(chinese) from tb_demo071),(select avg(math) from tb_demo071),(select avg(english) from tb_demo071) from
tb_demo071";
		$rsss = mysql_query($sql) or die ("false");                        //返回结果集
		$rsst = mysql_fetch_array($rsss);
?>
	<tr><td background="pic/head.JPG">各科平均成绩</td><td background="pic/head.JPG"></td><td background="pic/head.JPG"></td><td
background="pic/head.JPG"></td><td background="pic/head.JPG"><?php echo $rsst[0];?></td><td background="pic/head.JPG"><?php echo
$rsst[1];?></td><td background="pic/head.JPG"><?php echo $rsst[2];?></td></tr>
<?php
	}
?>
```

秘笈心法

心法领悟 416：方便的别名。

在使用子查询时最好为列表项取个别名，这样可以方便用户在使用 mysql_fetch_array()函数时为表项赋值。

实例 417 使用子查询关联数据

光盘位置：光盘\MR\08\417

高级

趣味指数：★★★★

实例说明

使用子查询关联数据在实际项目开发过程中被广泛应用，它可以使多表之间的查询变得更加简捷。运行本实例，如图 8.89 所示，首先在图中的文本框中输入要查询的班级名称，然后单击“查询”按钮，即可将该班级所有学生的详细信息查询出来。

使用子查询关联数据

输入查询班级 查询

学生ID	学生姓名	学生性别	语文成绩	数学成绩	英语成绩
3	小安	男	92	94	93
4	小巩	男	86	87	99

图 8.89　使用子查询关联数据

关键技术

本实例实现使用子查询关联数据的 SQL 语句的代码如下：

```
select * from tb_demo072_student where uid = (select id from tb_demo072_class where className = '$_POST[text]' );
```

设计过程

创建脚本文件 index.php。开启 MySQL 服务并连接 MySQL 数据库，编写文本框，为文本框设置初始值，当单击“查询”按钮时，显示如图 8.89 所示的内容，核心代码如下：

```
<?php
if($_POST[sub]){                                                              //单击按钮
        $rs = mysql_query("select * from tb_demo072_student where uid = (select id from tb_demo072_class where className = '$_POST[text]' )");
                                                                              //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">学生 ID</td><td background="pic/head.JPG">学生姓名</td><td
background="pic/head.JPG">学生性别</td><td background="pic/head.JPG">语文成绩</td><td background="pic/head.JPG">数学成绩</td><td
background="pic/head.JPG">英语成绩</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                   //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[2];?></td><td
background="pic/head.JPG"><?php echo $rst[3];?></td><td background="pic/head.JPG"><?php echo $rst[4];?></td><td
background="pic/head.JPG"><?php echo $rst[5];?></td><td background="pic/head.JPG"><?php echo $rst[6];?></td><td
background="pic/head.JPG"><?php echo $rst[7];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 417：浅谈本实例的实现原理。

本实例的实现原理是通过内查询 select id from tb_demo072_class where className = '$_POST[text]'获取要查询的班级名称，然后通过外部查询获取要查询的班级的所有学生的详细信息。



实例 418　多表联合查询

光盘位置：光盘\MR\08\418

高级

趣味指数：★★★★

实例说明

多表联合查询可以解决连接或简单子查询难于处理的问题。运行本实例，如图 8.90 所示，单击图中的“合并学生信息”按钮，即可将计算机系和数学系的学生信息合并输出。

多表联合查询

合并学生信息

学生ID	学生姓名	入学时间
1	小杨	2010-07-26
2	小刘	2010-07-20
3	小潘	2010-06-19
4	小安	2010-06-18
3	小雷	2010-07-26
4	小新	2010-07-17

图 8.90　多表联合查询

关键技术

利用 SQL 语言中的关键字 union，可以将不同表中符合条件的数据信息显示在同一列中。union 语句的使用格式如下：

```
<query specification1> [,query specification2……] union [all] <query specification1> [,query specification2……]
```

参数说明：

<query specification1>：是查询规范或查询表达式。

union：组合多个结果集并将其作为单个结果集返回。

all：在结果集中包含所有的行，包含重复行。

设计过程

创建脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。当单击“合并学生信息”按钮时，显示如图 8.90 所示的内容，核心代码如下：

```
<?php
 if($_GET[id] == "1"){                                                   //如果地址栏 id 等于 1
                                                                         //返回结果集
        $rs = mysql_query("select * from tb_demo074_student union select * from tb_demo074_fasten");
?>
<table width="580px"><tr><td background="pic/head.JPG">学生 ID</td><td background="pic/head.JPG">学生姓名</td><td
background="pic/head.JPG">入学时间</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                              //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 418：使用 union 关键字的注意事项。

在使用关键字 union 时应注意以下两点：

（1）在使用 union 运算符组合的语句中，所有选择列表的表达式数目必须相同，如列名、算术表达式及聚合函数等。

（2）在每个查询表中，对应列的数据必须是同一数据类型。

实例 419 对联合查询后的结果进行排序

光盘位置：光盘\MR\08\419

高级

趣味指数：★★★★

实例说明

为了与 union 运算兼容，要求所有 select 语句都不能有 order by 子句，但有一种情况例外，那就是在最后一个 select 语句中放置 order by 子句实现结果的最终排序输出。运行本实例，如图 8.91 所示，单击图中的“合并学生信息”按钮，即可将所有的学生信息合并，并按入学时间降序输出。

对联合查询后的结果进行排序

合并学生信息

学生ID	学生姓名	入学时间
1	小杨	2010-07-26
3	小雷	2010-07-26
2	小刘	2010-07-20
4	小新	2010-07-17
3	小潘	2010-06-19
4	小安	2010-06-18

图 8.91　对联合查询结果进行排序

关键技术

本实例实现对联合查询结果进行排序的 SQL 语句的代码如下：

```
select * from tb_demo074_student union select * from tb_demo074_fasten order by date desc;
```

设计过程

创建脚本文件 index.php。开启 MySQL 服务并连接 MySQL 数据库，当单击“合并学生信息”按钮时，显示如图 8.91 所示的内容，核心代码如下：

```
<?php
if($_GET[id] == "1"){                                                                    //如果地址栏参数 id 等于 1.
                                                                                         //返回结果集
        $rs = mysql_query("select * from tb_demo074_student union select * from tb_demo074_fasten order by date desc");
?>
<table width="580px"><tr><td background="pic/head.JPG">学生 ID</td><td background="pic/head.JPG">学生姓名</td><td
background="pic/head.JPG">入学时间</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                              //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 419：使用 union 关键字的注意事项。

union 语句在使用条件上相对比较苛刻，所以在使用此语句时一定要注意两个表项数目或字段类型是否相同。

实例 420　条件联合语句

光盘位置：光盘\MR\08\420

高级

趣味指数：★★★★

实例说明

本实例将应用 group by 分组语句和 having 语句实现条件联合查询。运行本实例，如图 8.92 所示，单击图中的“出版社信息”按钮，即可将全国各个大型出版社的名单显示出来，并保证人民邮电出版社和机械工业出版社始终位于名单最前列。

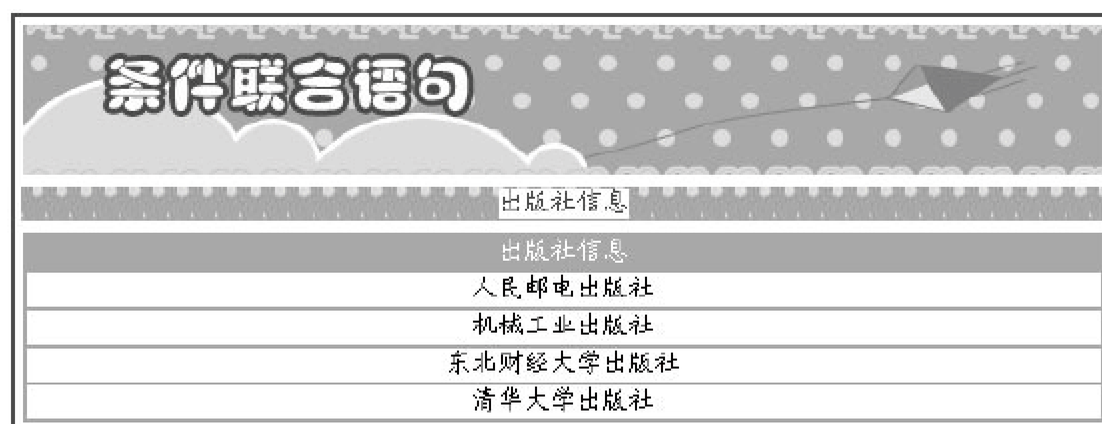


图 8.92　条件联合语句

关键技术

本实例的条件联合语句中，首先按出版社名称（name）进行分组，然后利用 having 子句限定条件，如 having name='人民邮电出版社' or name='机械工业出版社'，这样，执行结果将显示出人民邮电出版社和机械工业出版社这两条记录，最后用 union all 运算符与另一个数据表合并，并利用 having 限定条件（having name<>'人民邮电出版社' and name<>'机械工业出版社'），即可调出全国各大型出版社的名称，并且确保人民邮电出版社和机械工业出版社始终位于名单最前列。本实例中实现联合查询的代码如下：

```
select name from tb_demo076_Beijing group by name having name='人民邮电出版社' or name='机械工业出版社' union all select name from
tb_demo076_Beijing group by name having name<>'人民邮电出版社' and name<>'机械工业出版社'
```

设计过程

编写文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。当单击“出版社信息”按钮时，显示如图 8.92 所示的内容。本实例的核心代码如下：

```
<?php
if($_GET[id] == "1"){                                                                                    //单击按钮
        $rs = mysql_query("select name from tb_demo076_Beijing group by name having name='人民邮电出版社' or name='机械工业出版社' union all
select name from tb_demo076_Beijing group by name having name<>'人民邮电出版社' and name<>'机械工业出版社'");   //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG" align="center">出版社信息</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                                              //循环输出结果
?>
<tr><td background="pic/head.JPG" align="center"><?php echo $rst[0];?></td></tr>
<?php
        }
}
?>
```



秘笈心法

心法领悟 421：多表查询实现的方法。

本实例使用的内连接只是 SQL 语句的一种方法，此方法可以使用多表查询的知识实现相同功能。使用多表查询的 SQL 语句的代码如下：

```
select a.name,a.address,a.date,b.math,b.english,b.chinese from tb_demo065 as a,tb_demo065_tel as b where a.id=b.id;
```

实例 422 复杂内连接查询 高级

光盘位置：光盘\MR\08\422 趣味指数：★★★★

实例说明

在实际项目开发过程中，所应用的内连接查询大多都是较复杂的，例如对多张表通过内连接方式建立关联后，还需要在 SQL 语句中再指明一些查询条件，从而进一步实现查询的精确定位。运行本实例，如图 8.94 所示，首先在图中的文本框中输入要查询的员工编号，然后单击“查询”按钮，即可将所要查询的员工的基本信息和工资状况查询出来。

复杂内连接查询

输入员工编号 查询

员工姓名	员工性别	入职时间	员工工资
小杨	男	2010-07-26	2800

图 8.94 复杂的内连接查询

关键技术

复杂的内连接查询是在基本内连接查询的基础上再附加一些查询条件，例如本实例中实现查询员工基本信息情况的代码如下：

```
select tb_demo078_wages.Wages,tb_demo078_informaction.name,tb_demo078_informaction.sex,tb_demo078_informaction.date from
tb_demo078_wages inner join tb_demo078_informaction on tb_demo078_wages.id=tb_demo078_informaction.id where
tb_demo078_informaction.id=(select id from tb_demo078_id where code='$_POST[text]');
```

设计过程

创建脚本文件 index.php，开启 MySQL 服务，连接 MySQL 数据库。编写<form>表单，当单击“查询”按钮时，显示如图 8.94 所示的内容，核心代码如下：

```
<?php
if($_POST[sub]){                                                                    //单击按钮
        $rs = mysql_query("select tb_demo078_wages.Wages,tb_demo078_informaction.name,tb_demo078_informaction.sex,tb_demo078_informaction
.date from tb_demo078_wages inner join tb_demo078_informaction on tb_demo078_wages.id=tb_demo078_informaction.id where tb_demo078_
informaction.id=(select id from tb_demo078_id where code='$_POST[text]')");         //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG" align="center">员工姓名</td><td background="pic/head.JPG" align="center">员工性别
</td><td background="pic/head.JPG" align="center">入职时间</td><td background="pic/head.JPG" align="center">员工工资</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                         //循环输出结果
?>
<tr><td background="pic/head.JPG" align="center"><?php echo $rst[1];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[2];?></td><td background="pic/head.JPG" align="center"><?php echo $rst[3];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[0];?></td></tr>
<?php
```

```
        }
}
?>
```

秘笈心法

心法领悟 422：如何选择 SQL 语句查询的方法。

实现复杂 SQL 语句查询的方法有很多，而且彼此实现的功能相同，至于选择哪种方法，则由用户或程序员的个人喜好决定。

实例 423	两表的内连接关联 光盘位置：光盘\MR\08\423	高级 趣味指数：★★★★

实例说明

建立表与表之间的关联是 PHP 程序员在设计数据库时都会遇到的问题，而两表之间的关联在中小型网站中出现的频率是最高的。运行本实例，在图 8.95 所示界面的文本框中输入员工工资，单击“查询”按钮，即可显示与文本框关联的所有内容。

图 8.95　两表的内连接关联

关键技术

本实例实现两表关联的 SQL 语句的代码如下：

```
select tb_demo078_wages.Wages,tb_demo078_informaction.name,tb_demo078_informaction.sex,tb_demo078_informaction.date from tb_demo078_
wages inner join tb_demo078_informaction on tb_demo078_wages.id=tb_demo078_informaction.id where tb_demo078_informaction.id=(select id from
tb_demo078_id where code='$_POST[text]');
```

设计过程

编写脚本文件 index.php，开启 MySQL 服务，连接 MySQL 数据库，当单击“查询”按钮时，显示如图 8.95 所示的内容，核心代码如下：

```
if($_POST[sub]){                                                                    //单击按钮
        $rs = mysql_query("select tb_demo078_wages.Wages,tb_demo078_informaction.name,tb_demo078_informaction.sex,tb_demo078_informaction
.date from tb_demo078_wages inner join tb_demo078_informaction on tb_demo078_wages.id=tb_demo078_informaction.id where tb_demo078_
informaction.id=(select id from tb_demo078_id where code='$_POST[text]')");          //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG" align="center">员工姓名</td><td background="pic/head.JPG" align="center">员工性别
</td><td background="pic/head.JPG" align="center">入职时间</td><td background="pic/head.JPG" align="center">员工工资</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                         //循环输出结果
?>
<tr><td background="pic/head.JPG" align="center"><?php echo $rst[1];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[2];?></td><td background="pic/head.JPG" align="center"><?php echo $rst[3];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[0];?></td></tr>
<?php
        }
}
?>
```



left outer join 表示建立表 tb_yg 和 tb_yg_info 的外连接。

实例 425 left outer join 查询

光盘位置：光盘\MR\08\425

高级

趣味指数：★★★★

实例说明

如果要包含连接的两个表中的不匹配行，可以通过左连接或右连接的方式实现。运行本实例，如图 8.97 所示，单击图中的“查询”按钮，即可将员工信息表中的所有记录以及员工工资表中与员工 ID 相匹配的记录显示出来。

left outer join查询

查询

员工姓名	员工性别	入职时间	员工工资
2800	1	小杨	2010-07-26
2900	2	小潘	2010-07-26
3000	3	小刘	2010-07-26

图 8.97 left outer join 查询

关键技术

左连接返回的查询结果包含左表中的所有符合查询条件及右表中所有满足连接条件的行，MySQL 数据库中使用左连接的语法格式如下：

```
select field 1[field2…] from table1 left [outer] join table2 on join_condition [where search_condition]
```

参数说明：

left outer join：表示表之间通过左连接方式相互连接，也可以简写成 left join。

on join_condition：指多表建立连接所使用的连接条件。

where search_condition：可选项，用于设置查询条件。

设计过程

编写脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。当单击“查询”按钮时，显示如图 8.97 所示的内容，核心代码如下：

```
if($_GET[id] == "1"){                                          //单击按钮
        $rs = mysql_query("select * from tb_demo078_informaction left outer join tb_demo078_wages on tb_demo078_wages.id=
tb_demo078_informaction.id ");                                 //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG" align="center">员工姓名</td><td background="pic/head.JPG" align="center">员工性别
</td><td background="pic/head.JPG" align="center">入职时间</td><td background="pic/head.JPG" align="center">员工工资</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                     //循环输出结果
?>
<tr><td background="pic/head.JPG" align="center"><?php echo $rst[1];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[2];?></td><td background="pic/head.JPG" align="center"><?php echo $rst[3];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[5];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 425：对左连接的理解。

左连接以 left outer join 左侧的表名为基准，故称左连接，左侧表名的所有信息将被全部输出，符合条件的右侧表信息也会被输出，如果不符合条件，返回 NULL。

实例 426　right outer join 查询

光盘位置：光盘\MR\08\426

高级

趣味指数：★★★★

实例说明

通过右连接可以将右表中满足查询条件的记录全部显示出来，并显示左表中满足连接条件的记录。运行本实例，如图 8.98 所示，单击图中的“查询”按钮，即可将员工信息表中满足连接条件的记录及员工工资表中所有记录全部显示出来。

right outer join查询

查询

员工姓名	员工性别	入职时间	员工工资
小杨	男	小杨	2010-07-26
小潘	男	小潘	2010-07-26
小刘	男	小刘	2010-07-26

图 8.98　right outer join 查询

关键技术

右连接返回的查询结果包含左表中的所有符合连接条件以及右表中所有满足查询条件的行，MySQL 数据库中使用右连接的语法格式如下：

```
select field 1[field2…] from table1 right [outer] join table2 on join_condition [where search_condition]
```

参数说明：

right outer join：表示表之间通过右连接方式相互连接，也可以简写成 right join。

on join_condition：指多表建立连接所使用的连接条件。

where search_condition：可选项，用于设置查询条件。

设计过程

编写脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。当单击“查询”按钮时，显示如图 8.98 所示的内容。本实例的核心代码如下：

```
if($_GET[id] == "1"){                                                              //单击按钮
        $rs = mysql_query("select * from tb_demo078_wages right outer join tb_demo078_informaction on tb_demo078_wages.id=tb_demo078_
informaction.id ");                                                                 //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG" align="center">员工姓名</td><td background="pic/head.JPG" align="center">员工性别
</td><td background="pic/head.JPG" align="center">入职时间</td><td background="pic/head.JPG" align="center">员工工资</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                         //循环输出结果
?>
<tr><td background="pic/head.JPG" align="center"><?php echo $rst[3];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[4];?></td><td background="pic/head.JPG" align="center"><?php echo $rst[5];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[1];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 426：对右连接的理解。

右连接以 right outer join 右侧的表名为基准，故称右连接，右侧表名的所有信息将被全部输出，符合条件的左侧表信息也会被输出，如果不符合条件，返回 NULL。

实例 427 利用 in 或 notin 关键字限定范围

光盘位置：光盘\MR\08\427

高级

趣味指数：★★★★

实例说明

利用关键字 in 或 notin 可以限定查询范围，运行本实例，如图 8.99 所示，首先在图中的文本框中输入要查询学生成绩的学号范围，然后单击“查询”按钮，即可将该范围内的所有学生的各科成绩查找出来。

利用in或notin语句限定范围

请输入学号范围：　到　查询

学生编号	学生姓名	语文成绩	数学成绩	英语成绩
1081	小杨	99	98	97
1015	小刘	98	97	96
1026	小潘	94	95	96

图 8.99　利用 in 语句限定范围

关键技术

本实例中，利用关键字 in 查询指定学号范围内的学生成绩的代码如下：

```
select * from tb_demo083 where code in (select code from tb_demo083 where code between '$_POST[te]' and '$_POST[tex]');
```

设计过程

创建脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库。编写<form>表单，当单击“查询”按钮时，显示如图 8.99 所示的内容。核心代码如下：

```
if($_POST[sub]){                                                                                    //单击按钮
        $rs = mysql_query("select * from tb_demo083 where code in (select code from tb_demo083 where code between '$_POST[te]' and
'$_POST[tex]')");                                                                                   //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG" align="center">学生编号</td><td background="pic/head.JPG" align="center">学生姓名
</td><td background="pic/head.JPG" align="center">语文成绩</td><td background="pic/head.JPG" align="center">数学成绩</td><td
background="pic/head.JPG" align="center">英语成绩</td></tr>
<?php
        while($rst = mysql_fetch_row($rs)){                                                          //循环输出结果
?>
<tr><td background="pic/head.JPG" align="center"><?php echo $rst[1];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[2];?></td><td background="pic/head.JPG" align="center"><?php echo $rst[3];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[4];?></td><td background="pic/head.JPG" align="center"><?php echo $rst[5];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 427：如何书写不在指定范围的 SQL 语句代码。

在本实例的 SQL 语句代码中，如果想查找不在指定范围内的所有信息，可以将 in 改为 notin。

实例 428　用 in 查询表中的记录信息

光盘位置：光盘\MR\08\428

高级

趣味指数：★★★★

实例说明

SQL 语言中，可以利用关键字 in 来查询表中满足一定条件的记录。运行本实例，如图 8.100 所示，首先在图中的文本框中输入要查询的图书名称，然后单击“查询”按钮，即可将与用户输入的查询关键字相匹配的图书查找出来。

用IN查询表中的记录信息

输入关键字　查询

图书编号	图书名称	出版社	图书作者
1	《ASP.NET编程宝典》	机械工业出版社	明日科技
3	《PHP范例实战宝典》	清华大学出版社	明日科技

图 8.100　用 in 查询表中记录

关键技术

本实例首先通过 in 子查询查找出所有与用户输入的关键字相匹配的图书名称，然后通过外部查询查找出所有图书的详细信息，代码如下：

```
select * from tb_demo084 where name in (select name from tb_demo084 where name like '%$_POST[te]%';
```

设计过程

创建脚本文件 index.php。编写网页框架，开启 MySQL 服务，连接 MySQL 数据库。编写<form>表单，当单击“查询”按钮时，显示如图 8.100 所示的内容，核心代码如下：

```
<?php
if($_POST[sub]){                                                              //单击按钮
        if($_POST[te] == "" || $_POST[te] == "输入关键字"){                       //对文本框信息进行判断
                echo "<script>alert('请输入内容');</script>";                       //JavaScript 提示
        }else{
                $sql = "select * from tb_demo084 where name in (select name from tb_demo084 where name like '%$_POST[te]%' ) ";        //sql 语句
                $rs = mysql_query($sql);                                      //返回结果集
?>
                <table width="580px"><tr><td background="pic/head.JPG" align="center">图书编号</td><td background="pic/head.JPG"
align="center">图书名称</td><td background="pic/head.JPG" align="center">出版社</td><td background="pic/head.JPG" align="center">图书作者
</td></tr>
<?php
                while($rst = mysql_fetch_row($rs)){                           //循环输出结果
?>
                <tr><td background="pic/head.JPG" align="center"><?php echo $rst[0];?></td><td background="pic/head.JPG" align="center"><?php
echo $rst[1];?></td><td background="pic/head.JPG" align="center"><?php echo $rst[2];?></td><td background="pic/head.JPG" align="center"><?php
echo $rst[3];?></td></tr>
<?php
                }
        }
}
?>
```

秘笈心法

心法领悟 428：实现实例的其他方法。

在上面的实例中已经提到过，SQL 语句有很多种形式，而且不同的 SQL 语句形式可以完成相同的功能，本实例使用关键技术中的 SQL 语句未免有些麻烦，实现此功能的简单 SQL 语句代码如下：

```
select * from tb_demo084 where name like '%$_POST[te]%';
```

实例 429 由 in 引入的关联子查询

光盘位置：光盘\MR\08\429

高级

趣味指数：★★★★

实例说明

本实例实现利用 SQL 语言中的关键字 in 查找在职或离职的员工信息。运行本实例，如图 8.101 所示，首先在图中的单选按钮中选择员工的在岗情况，然后单击“查询”按钮，即可将所有在职或离职的员工信息显示出来。

由IN引入的关联子查询

◉在职　○离职　查询

员工ID	员工姓名	家庭住址	是否在职
1	小杨	吉林长春	在职
2	小潘	吉林长春	在职
3	小刘	吉林长春	在职

图 8.101　由 in 引入的关联子查询

关键技术

由关键字 in 引入的关联子查询的语法格式如下：

```
select <column name list> from <table list> where <test expression> in <subquery>
```

参数说明：

column name list：要查询的列名。

table list：查询所涉及的表名。

test expression：查询条件。

subquery：子查询。

本实例使用关键字 in 实现关联子查询的 SQL 语句的代码如下：

```
select * from tb_demo085 where work in (select work from tb_demo085 where work='$_POST[click]';
```

设计过程

创建脚本文件 index.php，开启 MySQL 服务，连接 MySQL 数据库。当单击“查询”按钮时显示如图 8.101 所示的内容，核心代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111");                                      //连接 MySQL
mysql_select_db("db_database08",$conn);                                               //连接 MySQL 数据库
mysql_query("SET NAMES UTF8");                                                        //设置编码集
if($_POST[sub]){                                                                      //单击按钮
                                                                                      //返回结果集
        $rs = mysql_query("select * from tb_demo085 where work in (select work from tb_demo085 where work='$_POST[click]')");
?>
<table width="580px"><tr><td background="pic/head.JPG" align="center">员工 ID</td><td background="pic/head.JPG" align="center">员工姓名
</td><td background="pic/head.JPG" align="center">家庭住址</td><td background="pic/head.JPG" align="center">是否在职</td></tr>
<?php
                while($rst = mysql_fetch_row($rs)){                                   //循环输出结果
?>
<tr><td background="pic/head.JPG" align="center"><?php echo $rst[0];?></td><td background="pic/head.JPG" align="center"><?php echo
$rst[1];?></td><td background="pic/head.JPG" align="center"><?php echo $rst[2];?></td><td background="pic/head.JPG" align="center"><?php echo
```









```
                <td  height="25" bgcolor="#FFFFFF"><div align="center"><?php $f=$rs->fields(syassort);echo
iconv("gbk","utf-8",$f->value);?></div></td>
                <td  height="25" bgcolor="#FFFFFF"><div align="center"><?php $f=$rs->fields(newdate);echo
iconv("gbk","utf-8",$f->value);?></div></td>
          </tr>
          <?php
             $rs->movenext;
           }
      }
    ?>
       </table></td>
    </tr>
</table>
<?php
  }
?>
```

秘笈心法

心法领悟 432：format()函数的局限性。

由于 format()函数只适用于 Access 数据库，在正常程序开发过程中，只有在使用 Access 数据库的程序中可以偶尔看见此函数。

实例 433　在查询中使用字符串函数

光盘位置：光盘\MR\08\433

高级

趣味指数：★★★★

实例说明

本实例通过使用字符串函数从学生信息表中查找 03 级全体学生的信息。运行本实例，如图 8.105 所示，单击图中的“查询”按钮即可将 03 级所有学生的信息显示出来。

在查询中使用字符串函数

明日科技

获取03级全体学生信息 查询

学号	姓名	班级	科目	成绩
0312101	小刘	三年一班	语文	10
0312102	小张	三年一班	英语	100
0313303	小王	三年三班	数学	20
0312104	小李	三年一班	几何	55
0312101	小刘	三年二班	代数	22
0312106	小潘	三年一班	物理	12

图 8.105　在查询中使用字符串函数查询 03 级学生信息

关键技术

获取某字段中指定位置的字符串，可以通过函数 mid()实现，该函数的使用格式如下：

```
mid(str,start [,length])
```

参数说明：

str：必要参数，字符串表达式，从中返回字符，如果 string 包含 null，将返回 null。

start：必要参数，为 long 型，是指 string 中被获取出部分的字符位置。如果 start 超过 string 的字符数，mid()函数将返回零长度字符。

length：可选参数，为 variant long，指要返回的字符数，如果省略或 length 超过文本的字符数（包括 start 处的字符），将返回字符串从 start 到尾端的所有字符。



在查询中使用日期函数

明日科技

根据生日查询员工年龄 查询

员工编号	姓名	性别	年龄	祖籍
yg001	小赵	男	26	吉林长春
yg002	小王	女	29	辽宁沈阳
yg003	小科	男	27	吉林四平

图 8.106　查询员工生日信息

关键技术

使用 DateDiff()函数可返回两个指定日期的时间间隔，该函数的语法格式如下：

```
DateDiff(interval ,date1, date2 [,firstday of week [,firstweek of year]])
```

参数说明：

interval：必选参数，字符串表达式，用来计算 date1 和 date2 的时间间隔。

data1,date2：必选参数，用于指定具体日期。

firstday of week：可选参数，指定一年的第一周的常数，如果未指定，则以包含 1 月 1 日的星期为第一周。

设计过程

（1）建立 conn.php 文件，实现与数据库的连接，代码如下：

```
<?php
$conn=new com("adodb.connection");                              //连接数据库
$connstr="provider=microsoft.jet.oledb.4.0;data source=".realpath("./data/db_database08.mdb");
$conn->open($connstr);
?>
```

（2）执行查询，并通过函数 DateDiff()计算所有员工的年龄，显示查询结果，代码如下：

```
        <?php
         include("conn.php");                                     //包含数据库连接文件
         $rs=new com("adodb.recordset");
        //查询结果
         $rs->open("select ygid,name,sex,address,DateDiff('yyyy',birthday,DATE()) as age from tb_worker",$conn,3,1);
         if($rs->eof|| $rs->bof) {
         ?>
        <tr>
          <td height="25" colspan="8" bgcolor="#FFFFFF"><div align="center"></div></td>
        </tr>
        <?php
          }else{
             while(!$rs->eof){
        ?>
        <tr>

        <td   height="25" background="../089/images/head.JPG"><div align="center"><?php $f=$rs->fields("ygid"); echo
iconv('gbk','utf-8',$f->value);?></div></td>
          <td   height="25" background="../089/images/head.JPG"><div align="center"><?php $f=$rs->fields("name"); echo
iconv('gbk','utf-8',$f->value);?></div></td>
            <td   height="25" background="../089/images/head.JPG"><div align="center"><?php $f=$rs->fields("sex"); echo
iconv('gbk','utf-8',$f->value); ?></div></td>
            <td   height="25" background="../089/images/head.JPG"><div align="center"><?php $f=$rs->fields("age"); echo
iconv('gbk','utf-8',$f->value);?></div></td>
            <td   height="25" background="../089/images/head.JPG"><div align="center"><?php $f=$rs->fields("address"); echo
iconv('gbk','utf-8',$f->value);?></div></td>
        </tr>
        <?php
           $rs->movenext;
          }
     }
   ?>
     </table></td>
```

```
  </tr>
</table>
<?php
 }
?>
```

秘笈心法

心法领悟 434：对 DateDiff()函数的参数的解析。

DateDiff()函数中的参数 interval 用来计算 date1 和 date2 的时间间隔。此参数可以选择一些设定数，例如 yyyy、q、m、y 等。分别表示年、季、月、一年的日数等。

实例 435　利用 having 语句过滤分组数据

光盘位置：光盘\MR\08\435

高级

趣味指数：★★★★

实例说明

本实例将统计员工信息表中员工工资大于 1500 元的员工职称。运行本实例，如图 8.107 所示，单击图中的“查询”按钮，即可将所有工资在 1500 元以上的员工职称显示出来。

利用having语句过滤分组数据

查询

员工ID	员工姓名	员工职称	员工工资
1	小杨	程序员	2000
2	小潘	程序员	2500
3	小刘	部门经理	3000

图 8.107　查询工资大于 1500 元的员工职称

关键技术

having 语句的语法格式如下：

```
[having <search_condition>]
```

参数<search_condition>指定组或聚合应满足的条件。当 having 与 group by all 一起使用时，having 子句可替代 all。

having search_condition 语句与 group by 语句合用，可用来设置一些被含入查询结果的“组别”所要符合的条件。having 语句可以包含多个查询所需要的条件，并且这些过滤条件之间通过 and 或 or 运算符相连接，同时也可以利用 not 运算符来逆转一个布尔表达式。

设计过程

创建脚本文件 index.php，开启 MySQL 服务并连接 MySQL 数据库，当单击“查询”按钮时，显示如图 8.107 所示的内容，核心代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die ("Connect MySQL False");        //连接 MySQL
mysql_select_db("db_database08",$conn);                                               //连接 MySQL 数据库
mysql_query("SET NAMES utf8");                                                        //设置编码格式
if($_GET[id] == "1"){                                                                 //当地址栏参数等于 1 时
        $rs = mysql_query("select * from tb_demo091 group by id having sal>1500") or die ("false");   //返回结果集
?>
<table width="580px"><tr><td background="pic/head.JPG">员工 ID</td><td background="pic/head.JPG">员工姓名</td><td
background="pic/head.JPG">员工职称</td><td background="pic/head.JPG">员工工资</td></tr>
<?php
```

```
        while($rst = mysql_fetch_row($rs)){                                        //循环输出结果
?>
<tr><td background="pic/head.JPG"><?php echo $rst[0];?></td><td background="pic/head.JPG"><?php echo $rst[1];?></td><td
background="pic/head.JPG"><?php echo $rst[2];?></td><td background="pic/head.JPG"><?php echo $rst[3];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 435：如何使用 having 子句。

having 子句用于指定组或聚合的搜索条件。having 通常与 group by 子句一起使用，如果 SQL 语句中不含 group by 子句，having 的行为与 where 子句一样。

8.8　数据的导入和导出

在对网站的数据库进行管理的过程中，经常需要将数据库中的数据从一个数据库导入到另一个数据库中，有时还涉及到不同数据库之间的导入和导出。

实例 436	MySQL 数据自身的导入和导出 光盘位置：光盘\MR\08\436	高级 趣味指数：★★★★

实例说明

使用 PHP 语言开发与数据库操作有关的程序时，最佳搭档是 MySQL 数据库，要在 MySQL 数据库中实现导入和导出的操作方法有很多，这里介绍一种通过程序来实现导入和导出的方法。运行本实例，如图 8.108 所示。



图 8.108　数据自身的导入和导出

关键技术

实现数据导入和导出的原理是：首先从 A 数据库中读取要导出的数据，并将该数据输出到页面中。然后与 B 数据库建立连接，将从 A 数据库中读取到的数据存储到 B 数据库指定的表中。

设计过程

（1）新建 conn.php 文件，连接 MySQL 数据库，设置页面编码格式为 utf-8，其代码如下：

```
<?php
$conn = mysql_connect("localhost","root","111") or die ("Connect MySQL False");          //连接数据库服务器
mysql_select_db("db_database08",$conn) or die ("Connect Database False");               //连接数据库
mysql_query("SET NAMES UTF8");              //设置数据库编码格式
?>
```





图 8.109　导出 SQL Server 数据



图 8.110　创建 MySQL 数据表名称

关键技术

本实例需要连接 SQL Server 数据库和 MySQL 数据库，然后从 SQL Server 数据库的指定表中读取数据，将数据添加到 MySQL 数据库的指定表中。连接 MySQL 和 SQL Server 数据库的方法如下：

```
<?php
$conn = mssql_connect("PC-201006101638","sa","") or die ("Connect MsSQL False");    //连接 SQL Server
mssql_select_db("db_database0008",$conn) or die ("Connect Database False");         //连接数据库.
$con = mysql_connect("localhost","root","111") or die ("Connect MySQL False");       //连接 MySQL
mysql_select_db("db_database08",$con) or die ("Connect Database False");             //连接数据库
mysql  query("SET NAMES UTF8");
?>
```

设计过程

（1）新建 conn.php 文件，分别连接 MySQL 和 SQL Server 数据库。

（2）新建 index.php 文件。首先，通过 mssql 函数库操作 SQL Server 数据库，查询并输出 SQL Server 指定数据表中的数据。然后在 MySQL 数据库中创建一个指定的数据表，并且将从 SQL Server 数据库中读取的数据添加到 MySQL 数据库的数据表中，其代码如下：

```
?php
if($_GET[id] == "1"){                                                  //单击按钮
    include('conn.php');                                               //包含数据库连接文件
    $ms = mssql_query("select * from tb_demo093");                     //查询数据库操作
?>
<table width="580px"><tr><td>SQL Server 数据信息</td></tr><tr><td background="pic/head.JPG"><?php echo
mssql_field_name($ms,0);?></td><td background="pic/head.JPG"><?php echo mssql_field_name($ms,1);?></td><td
background="pic/head.JPG"><?php echo mssql  field  name($ms,2);?></td><td background="pic/head.JPG"><?php echo
mssql_field_name($ms,3);?></td><td background="pic/head.JPG"><?php echo mssql_field_name($ms,4);?></td><td
background="pic/head.JPG"><?php echo mssql_field_name($ms,5);?></td></tr>
<?php
    while($mms = mssql_fetch_array($ms)){                              //循环输出结果
?>

<tr><td background="pic/head.JPG"><?php echo $mms[id];?></td><td background="pic/head.JPG"><?php echo
iconv("gbk","utf-8",$mms[name]);?></td><td background="pic/head.JPG"><?php echo $mms[code];?></td><td background="pic/head.JPG"><?php
echo $mms[dept];?></td><td background="pic/head.JPG"><?php echo $mms[sal];?></td><td background="pic/head.JPG"><?php echo
iconv("gbk","utf-8",$mms[date]);?></td></tr>
<?php
    }
?>
if($_GET[id] == "2"){                                                  //单击按钮
    include("conn.php");                                               //包含数据库连接页
    $rs = mysql_query("select * from tb_demo093");                     //查询数据库操作
?>
<table class="tee" width="580px"><tr><td class="th">MySQL 数据信息</td></tr><tr><td background="pic/head.JPG"><?php echo "员工
ID\\".mysql_field_name($rs,0);?></td><td background="pic/head.JPG"><?php echo "员工姓名\\".mysql_field_name($rs,1);?></td><td
background="pic/head.JPG"><?php echo "员工编号\\".mysql_field_name($rs,2);?></td><td background="pic/head.JPG"><?php echo "所属部门
\\".mysql_field_name($rs,3);?></td><td background="pic/head.JPG"><?php echo "员工工资\\".mysql_field_name($rs,4);?></td><td
background="pic/head.JPG"><?php echo "当前时间\\".mysql_field_name($rs,5);?></td></tr>
<?php
    while($rst = mysql_fetch_array($rs)){                              //循环输出结果
```

```
?>
<tr><td background="pic/head.JPG"><?php echo $rst[id];?></td><td background="pic/head.JPG"><?php echo $rst[name];?></td><td
background="pic/head.JPG"><?php echo $rst[code];?></td><td background="pic/head.JPG"><?php echo $rst[dept];?></td><td
background="pic/head.JPG"><?php echo $rst[sal];?></td><td background="pic/head.JPG"><?php echo $rst[date];?></td></tr>
<?php
        }
}
?>
```

秘笈心法

心法领悟 437：连接 SQL Server 数据库的注意事项。

PHP 在默认情况下，是不能连接 SQL Server 数据库的。为了能够连接 SQL Server 数据库，必须修改 php.ini 文件的设置（在更改文件前需要去除该文件的只读属性），将;extension=php_mssql.dll 语句前的分号删除，然后重新启动 Apache 服务器即可。

实例 438　Access 数据导入到 MySQL 数据库中

光盘位置：光盘\MR\08\438

高级

趣味指数：★★★★

实例说明

在对数据库进行管理的过程中，不同数据库之间的数据转换是很麻烦的，本实例利用 ADO 连接 Access 数据库，将 Access 数据库中的数据导入到 MySQL 数据库中，运行结果如图 8.111 所示。

Access数据导入到MySQL数据库中

导出Access数据

MySQL数据信息

id	name	dept	sal	date
1	小杨	PHP	2000	2010-07-29
2	小刘	PHP	3000	2010-07-29
3	小潘	PHP	2500	2010-07-29

图 8.111　显示导入到 MySQL 的数据

关键技术

本实例重点讲解如何将 PHP 与 Aceess 数据库建立连接，及如何使用 PHP 语句从 Aceess 数据库中读取数据，并将数据添加到 MySQL 数据库中。

这里使用微软的 ADODB 数据库驱动实现 PHP 与 Aceess 数据库的连接。PHP 是通过预定义类 com 来使用 ADO 方法操纵 Access 数据库的，该类的详细说明如下：

```
string com::com( string module_name [, string server_name [, int codepage]])
```

参数说明：

module_name：被请求组件的名字或 class-id。

server_name：DCOM 服务器的名字。

Codepage：指定用于将 PHP 字符串转换成 UNICODE 字符串的代码页，反之亦然。该参数的取值有 CP_ACP、CP_MACCP、CP_OEMCP、CP_SYMBOL、CP_THREAD_ACP、CP_UTF7 和 CP_UTF8。

PHP 利用 com 类并使用 ADO 方法访问数据库的代码如下：

```
<?php
$conn=new com("adodb.connection");
$connstr="provider=microsoft.jet.oledb.4.0;data source=".realpath("./data/score.mdb");
$conn->open($connstr);
?>
```

设计过程

（1）新建 conn.php 文件，完成与 Access 数据库和 MySQL 数据库的连接，其代码如下：

```
<?php
$conn=new com("adodb.connection");
$connstr="provider=microsoft.jet.oledb.4.0;data source=".realpath("Access database/db_demo094.mdb");
$conn->open($connstr);
$rs=new com("adodb.recordset");
$con = mysql_connect("localhost","root","111") or die ("Connect MySQL False");
mysql_select_db("db_database08",$con) or die ("Connect Database False");
mysql_query("SET NAMES UTF8");
?>
```

（2）新建 index.php 文件，首先查询 Access 数据库，完成 Access 数据库中数据的循环输出，其代码如下：

```
<?php
if($_GET[id] == "1"){                                                  //单击按钮
        include("conn.php");                                           //包含数据库连接文件
        $rs->open("select * from tb_demo094",$conn);                   //返回结果集
        if($rs->eof|| $rs->bof){}else{
?>
Access 数据信息
<?php
                while(!$rs->eof){                                      //循环输出结果
?>
<tr>
<td><?php $f=$rs->fields(id);echo iconv("gbk","utf-8",$f->value);?></td><td><?php $f=$rs->fields(name);echo iconv("gbk","utf-8",$f->value);?></td>
<td><?php $f=$rs->fields(dept);echo iconv("gbk","utf-8",$f->value);?></td><td><?php $f=$rs->fields(sal);echo iconv("gbk","utf-8",$f->value);?></td>
<td><?php $f=$rs->fields(date);echo iconv("gbk","utf-8",$f->value);?></td>
</tr>
<?php
                        $rs->movenext;
                }
        }
?>
        <form action="" method="post">
                <input type="text" name="te" value="创建导入 MySQL 数据表名称" onfocus="this.value=''" class="one" />
                <input type="submit" name="sub" value=" " class="two" />
        </form>
<?php
}
?>
```

（3）将 Access 数据库中的数据通过 SQL 语句插入到 MySQL 数据库中，并且循环输出 MySQL 数据库中的数据，其代码如下：

```
if($_GET[id] == "2"){                                          //单击按钮
        include("conn.php");                                   //包含数据库连接页
        $rs = mysql_query("select * from tb_demo094");         //返回结果集
?>
<table width="580px">
<tr>
<td class="th">MySQL 数据信息</td></tr>
<tr><td background="pic/head.JPG">id</td>
<td background="pic/head.JPG">name</td>
<td background="pic/head.JPG">dept</td>
<td background="pic/head.JPG">sal</td>
<td background="pic/head.JPG">date</td>
</tr>
<?php
        while($rst = mysql_fetch_array($rs)){                  //循环输出结果
?>
<tr>
<td background="pic/head.JPG"><?php echo $rst[id];?></td>
<td background="pic/head.JPG"><?php echo $rst[name];?></td>
<td background="pic/head.JPG"><?php echo $rst[dept];?></td>
<td background="pic/head.JPG"><?php echo $rst[sal];?></td>
<td background="pic/head.JPG"><?php echo $rst[date];?></td>
</tr>
<?php
```

```
        }
}
?>
```

秘笈心法

心法领悟 438：ActiveX Data Objects（ADO）。

ActiveX Data Objects（ADO）是 Microsoft 开放数据库应用程序的数据库访问技术，它被设计用来同新的数据访问层 OLE DB Provider 一起协同工作，提供通用数据访问（Universal Date Access）。OLE DB 是一个低层的数据访问接口，用它可以访问各种数据源，包括传统的关系型数据库、电子邮件系统及自定义的商业对象。ADO 技术大大简化了 OLE DB 的操作，因为 ADO 封装 OLE DB 程序中使用的大量 COM 接口，所以 ADO 是一种高层的访问技术。

8.9 MySQL 数据库的备份和恢复

MySQL 数据库的功能是非常强大的，为确保数据库中数据的安全，对数据定期进行备份是很重要的。

实例 439	通过命令模式备份数据库 光盘位置：光盘\MR\08\439	高级 趣味指数：★★★★

实例说明

在命令行模式下可以实现创建、修改、删除数据库、数据表及数据表中的数据信息的操作。同样也可以在命令模式下对数据库进行备份，可通过 Mysql dump 命令将整个数据库备份到指定文件夹下的文本文件中，其运行效果如图 8.112 和图 8.113 所示。



图 8.112　cmd 命令备份数据库

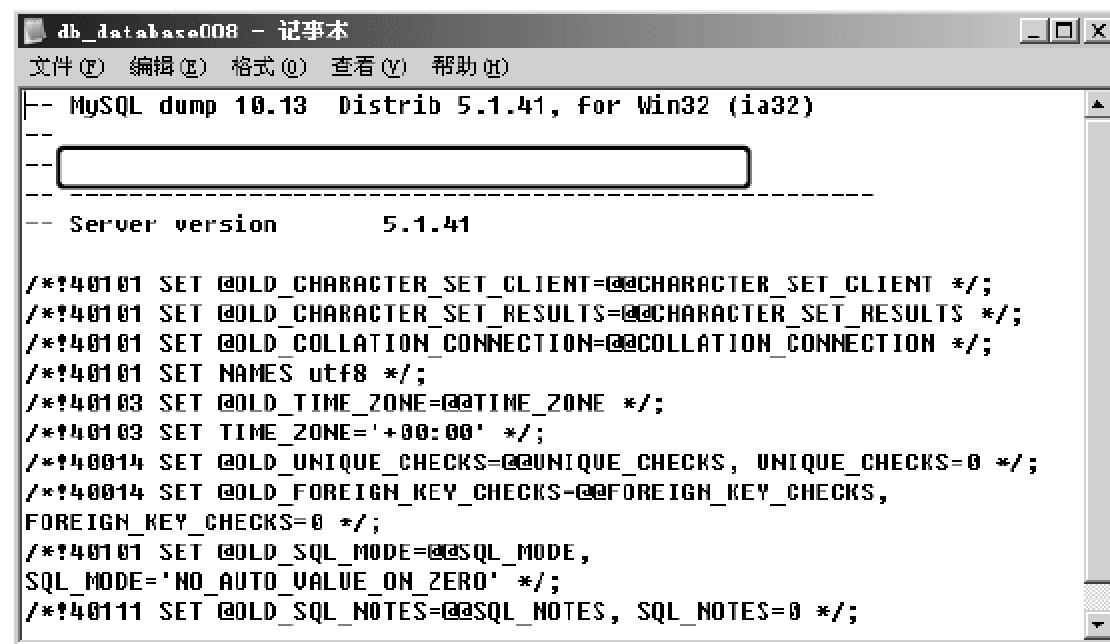


图 8.113　文本文件备份数据库内容

关键技术

在命令模式下实现对数据库的备份，应用的是 Mysql dump 命令，其语法如下：

```
Mysqldump –uroot –p111 db_database08 > D:\db_database08.txt;
```

设计过程

执行备份操作的步骤如下：

（1）选择“开始”菜单中的“运行”命令。

（2）在弹出的对话框的文本框中输入 cmd，单击“确定”按钮，进入命令模式。

（3）在命令模式中直接输入 mysqldump -uroot -p111 db_database08 > db_database08.txt，然后按 Enter 键即可。

其中，-uroot 中的 root 是用户名，而-p111 中的 111 是数据库密码；db_database08 是数据库名称；D:\db_database08.txt 是数据库备份的存储位置。

秘笈心法

心法领悟 439：手动设置环境变量。

在命令模式下操作 MySQL 数据库，最好先对计算机的环境变量进行设置。右键单击“我的电脑”图标，从弹出的快捷菜单中选择“属性”命令，在弹出的对话框中单击“高级”按钮，然后在新弹出的对话框中选择“环境变量”选项，并在用户变量的文本框中找到变量“path”并将其选中，单击“编辑”按钮，在变量 path 的变量值文本框中输入路径 F:\xampp\xampp\mysql\bin（MySQL 数据库中 bin 文件夹的安装路径），然后单击“确定”按钮。其中添加的 bin 文件夹的路径根据安装 MySQL 数据库的位置而定。

实例 440	通过 phpMyAdmin 备份数据库 光盘位置：光盘\MR\08\440	高级 趣味指数：★★★★

实例说明

备份 MySQL 数据库的操作可以在命令模式下实现，也可以通过 phpMyAdmin 图形化管理工具来实现。

phpMyAdmin 为 Web 开发人员提供类似 Access、SQL Server 的图形化数据库操作界面，通过该管理工具可以进行绝大部分的 MySQL 操作，包括对数据库及数据表的建立和维护。

关键技术

本实例运用 phpMyAdmin 图形化管理工具完成对数据库的备份操作。在备份数据库之前，首先选择需要备份的数据库，然后根据导航栏的提示实现数据库备份即可。

设计过程

通过 phpMyAdmin 图形化管理工具备份数据库，既可以备份整个数据库，也可以备份数据库中的一个表。

（1）打开 phpMyAdmin 图形化管理工具，在地址栏输入 http://127.0.0.1/phpMyAdmin/，进入到如图 8.114 所示的页面中。



图 8.114　phpMyAdmin 图形化管理工具的页面



秘笈心法

心法领悟 441：处理不同版本数据库的差异。

由于使用的 MySQL 数据库的版本不同，在进行备份时，有的版本需要停止 MySQL 服务后，才可以备份文件，以 XAMPP 为例，启动和停止 MySQL 服务的方法如图 8.117 所示。

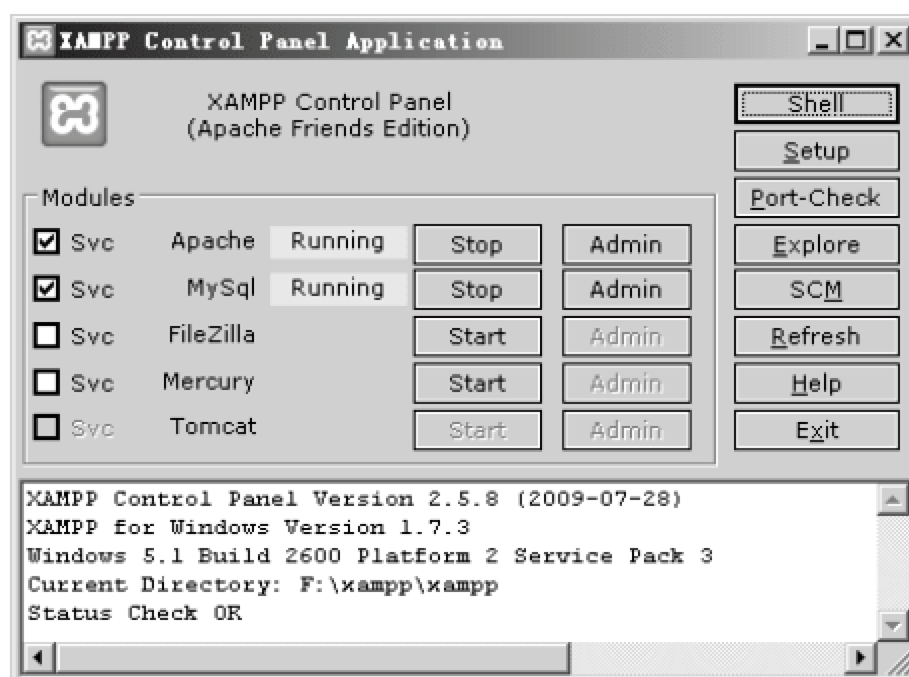


图 8.117　停止或启动 MySQL 服务

实例 442　**通过命令模式恢复数据库**　**高级**

光盘位置：光盘\MR\08\442　趣味指数：★★★★

实例说明

本实例将介绍通过命令模式恢复数据库的方法，它针对的是实例 439 中通过命令模式备份数据库的操作。

关键技术

通过命令模式恢复数据库使用的是 mysql -uroot -p111 db_database08 < D:\db_database08.txt 命令，其中-u 后的 root 代表用户名，-p 后的 root 代表密码，db_database08 代表数据库名称（或表名），“<”号后面的 D:\db_database08.txt 是数据库备份文件存储的位置。

设计过程

在命令模式下恢复整个数据库文件，命令如下：

```
C:\Documents and Setting\Administrator > mysql –uroot –p111db_database08 < D:\db_database08.txt;
```

该命令的操作步骤如下：

（1）选择“开始”菜单中的“运行”命令。

（2）在弹出的对话框中输入 cmd 命令，单击“确定”按钮，进入命令模式。

（3）在命令模式中直接输入 mysql -uroot -p111 db_database08 < D:\db_database08.txt，然后按 Enter 键，即可实现数据库的恢复，运行效果如图 8.118 所示。

秘笈心法

心法领悟 442：命令行下备份和恢复数据库的缺点。

由于此种备份和恢复数据库的方法都是要依托客户机硬盘上的文本文件，而文本文件是以明文的方式保存数据库信息的，所以安全性较低。

```
C:\WINDOWS\system32\cmd.exe
C:\Documents and Settings\Administrator>mysql -uroot -p111 db_database008 < D:\d
b_database008.txt;
mysql  Ver 14.14 Distrib 5.1.41, for Win32 (ia32)
Copyright 2000-2008 MySQL AB, 2008 Sun Microsystems, Inc.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL license
Usage: mysql [OPTIONS] [database]
  -?, --help          Display this help and exit.
  -I, --help          Synonym for -?
  --auto-rehash       Enable automatic rehashing. One doesn't need to use
                      'rehash' to get table and field completion, but startup
                      and reconnecting may take a longer time. Disable with
                      --disable-auto-rehash.
  -A, --no-auto-rehash
                      No automatic rehashing. One has to use 'rehash' to get
                      table and field completion. This gives a quicker start of
                      mysql and disables rehashing on reconnect. WARNING:
                      options deprecated; use --disable-auto-rehash instead.
  -B, --batch         Don't use history file. Disable interactive behavior.
                      (Enables --silent)
  --character-sets-dir=name
                      Directory where character sets are.
  --column-type-info  Display column type information.
  -c, --comments      Preserve comments. Send comments to the server. The
                      default is --skip-comments (discard comments), enable
                      with --comments
  -C, --compress      Use compression in server/client protocol.
  -#, --debug[-#]     This is a non-debug version. Catch this and exit
  --debug-check       Check memory and open file usage at exit.
  -T, --debug-info    Print some debug info at exit.
  -D, --database=name Database to use.
  --default-character-set=name
                      Set the default character set.
  --delimiter=name    Delimiter to be used.
  -e, --execute=name  Execute command and quit. (Disables --force and history
                      file)
  -E, --vertical      Print the output of a query (rows) vertically.
  -f, --force         Continue even if we get an sql error.
  -G, --named-commands
                      Enable named commands. Named commands mean this program's
                      internal commands; see mysql> help . When enabled, the
                      named commands can be used from any line of the query,
                      otherwise only from the first line, before an enter.
                      Disable with --disable-named-commands. This option is
```

图 8.118　命令行下恢复数据库数据

实例 443　通过 phpMyAdmin 恢复数据库

光盘位置：光盘\MR\08\443

高级

趣味指数：★★★★

实例说明

本实例介绍通过 phpMyAdmin 图形化管理工具恢复数据库的方法，它针对的是实例 440 中通过 phpMyAdmin 图形化管理工具备份数据库的操作。

关键技术

本实例的操作方法十分简单，相关技术已经在实例 440 中介绍过了，这里不再赘述。

在执行恢复数据库的操作时，首先必须在 phpMyAdmin 图形化管理工具中选中一个数据库，然后再执行导入文件的操作。

设计过程

通过 phpMyAdmin 图形化管理工具恢复数据库的步骤如下：

（1）打开 phpMyAdmin 图形化管理工具，在该界面左侧窗口的下拉菜单中选择要恢复的数据库。

（2）在 phpMyAdmin 图形化管理工具操作界面的左侧窗口中单击 SQL 查询窗口按钮，进入界面，在查询窗口中单击“导入文件”按钮，根据提示浏览文本文件所在的位置，找到该文件后，单击“执行”按钮即可。

秘笈心法

心法领悟 443：phpMyAdmin 图形化管理工具的版本差异。

phpMyAdmin 图形化管理工具因版本不同，实现导入文本文件的方法也不相同，但实现步骤大致相同。



了的操作界面，使程序员在设计 MySQL 数据库时变得简单。phpMyAdmin 几乎可以实现操作数据库的所有功能。本实例将演示如何通过 phpMyAdmin 修改 MySQL 用户密码。

关键技术

通过 phpMyAdmin 修改 MySQL 用户的密码可以通过两种方法实现：

（1）直接通过 phpMyAdmin 的界面进行更改。

（2）通过修改 phpMyAdmin 文件夹下的配置文件 config.php 实现更改。

设计过程

☑ 通过修改配置文件进行密码修改。

（1）将下载的 phpMyAdmin-x.zip 文件解压到 Apache 默认的根目录下，解压后的名称是 phpMyAdmin-2.x.x.x，其中的 2.x.x.x 是版本号，为方便使用，可以将其重新命名为 phpMyAdmin。

（2）打开 phpMyAdmin 文件夹，找到配置文件 config.php，然后使用记事本或其他编辑工具打开该文件，找到该文件中的以下内容进行配置：

```
$cfg['PmaAbsoluteUri'] = 'http://127.0.0.1/phpMyAdmin';
```

以上用于设置 phpMyAdmin 的 URL，例如 http://127.0.0.1/phpMyAdmin。

```
$cfg['blowfish_secret'] = 'root';
```

在上述语句中更改用户密码，例如 root。

```
$cfg['Servers'][$i]['controluser'] = 'root';
$cfg['Servers'][$i]['controlpass'] = 'root';
$cfg['Servers'][$i]['user'] = 'root';
$cfg['Servers'][$i]['password'] = 'root';
```

（3）完成配置后保存该文件。

☑ 通过 phpMyAdmin 的界面进行密码修改。

（1）登录到 phpMyAdmin 图形化工具页面，单击“修改密码” 按钮，如图 8.119 所示。

（2）根据页面提示输入新密码和确认密码，如图 8.120 所示。

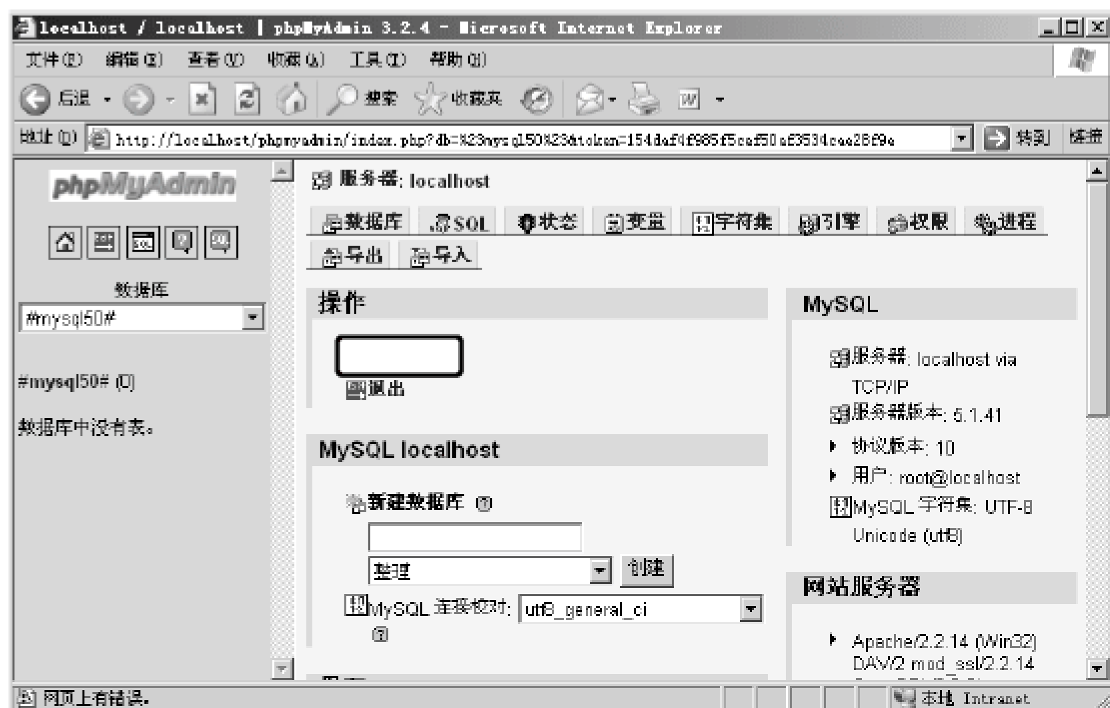


图 8.119　phpMyAdmin 图形化界面

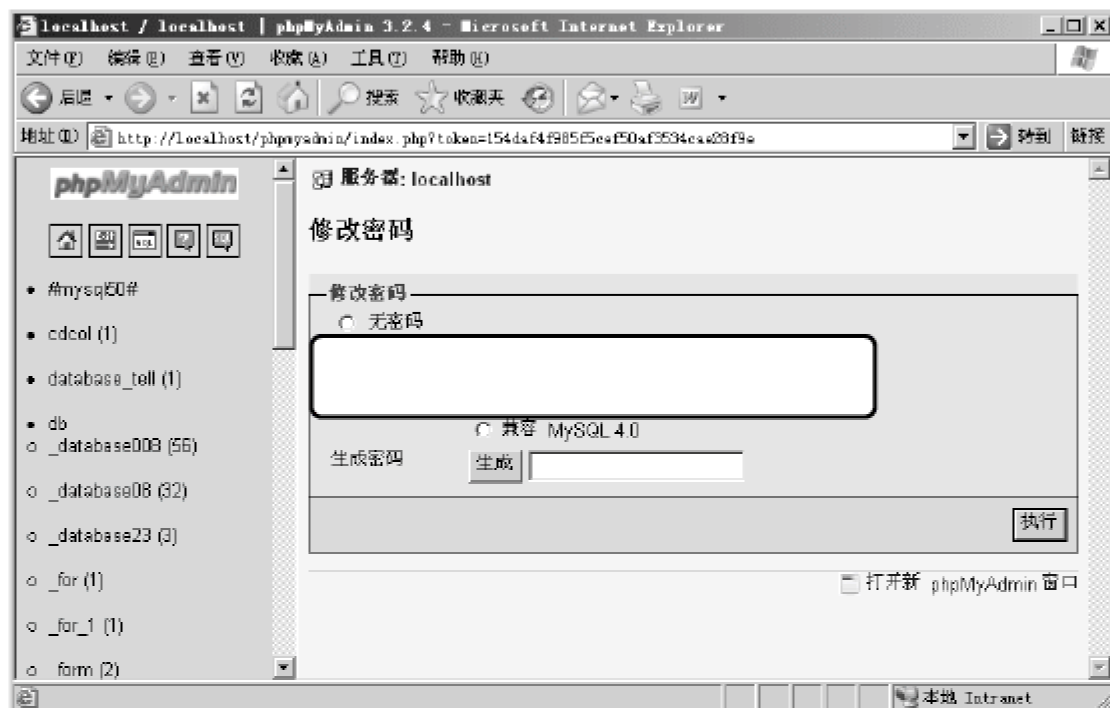


图 8.120　修改 MySQL 数据库密码

（3）单击“执行”按钮，修改 MySQL 数据库密码，如图 8.121 所示。

秘笈心法

心法领悟 445：phpMyAdmin 图形化管理工具的优点。

phpMyAdmin 图形化管理工具几乎是所有的 PHP+MySQL 用户必备的利器之一，其优点是：

（1）程序本身由 PHP 语言编写，并且是开源的，对于 PHP 编程爱好者来说，再适合不过。

（2）工具虽然免费，但功能并不欠缺，它可以很完美地完成对 MySQL 数据库的各种操作。

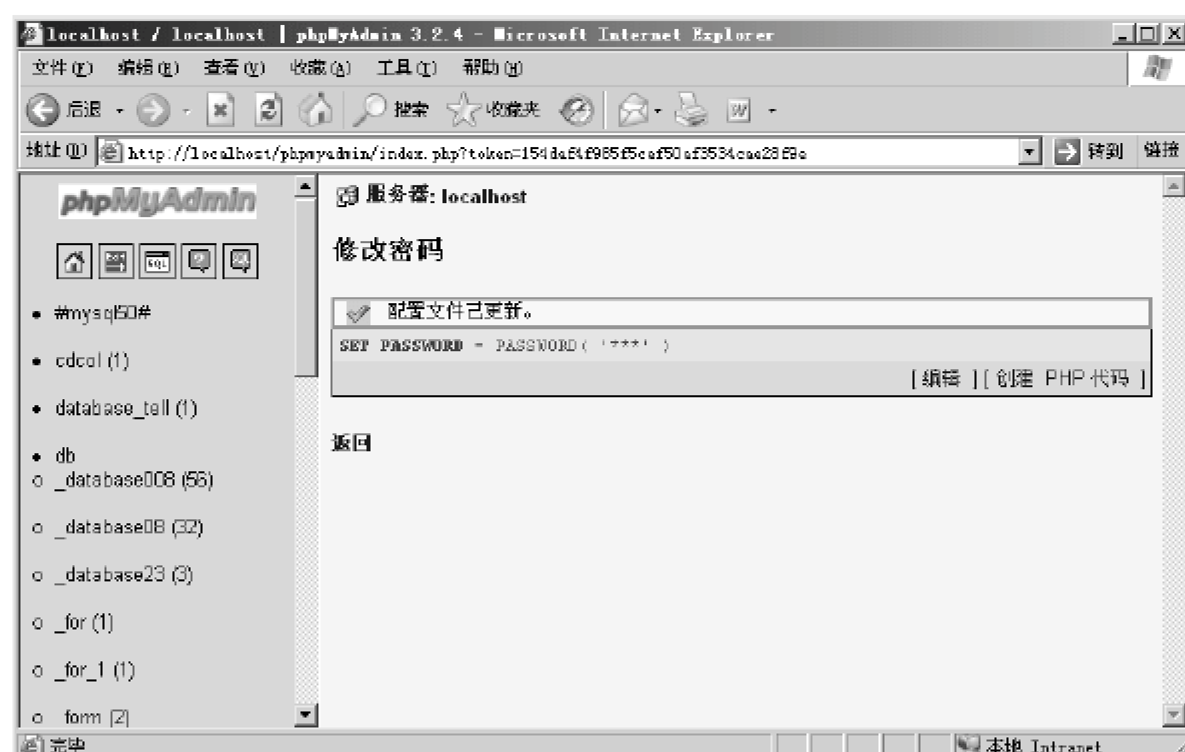


图 8.121　完成密码修改

实例 446　通过 phpMyAdmin 设置数据库、数据表编码　高级

光盘位置：光盘\MR\08\446　趣味指数：★★★★

实例说明

将页面、程序文件、数据库与数据表设置为统一的编码格式可以避免程序运行时出现乱码。一般情况下，设置页面的编码格式由 HTML 中的<meta>标签实现，设置程序文件的编码格式由 mysql_query()实现，设置数据库与数据表的编码格式通过 phpMyAdmin 实现。

关键技术

本实例的操作方法十分简单，在创建数据库名称时，在“整理”下拉列表中选择需要设置的编码格式；在创建数据表时，同样将数据字段设置为统一的编码格式。

设计过程

（1）登录到 phpMyAdmin 图形化工具页面，创建数据库名称，选择编码格式，如图 8.122 所示。

（2）创建数据表，定义数据表字段，并设置编码格式，如图 8.123 所示。



图 8.122　设置数据库编码格式

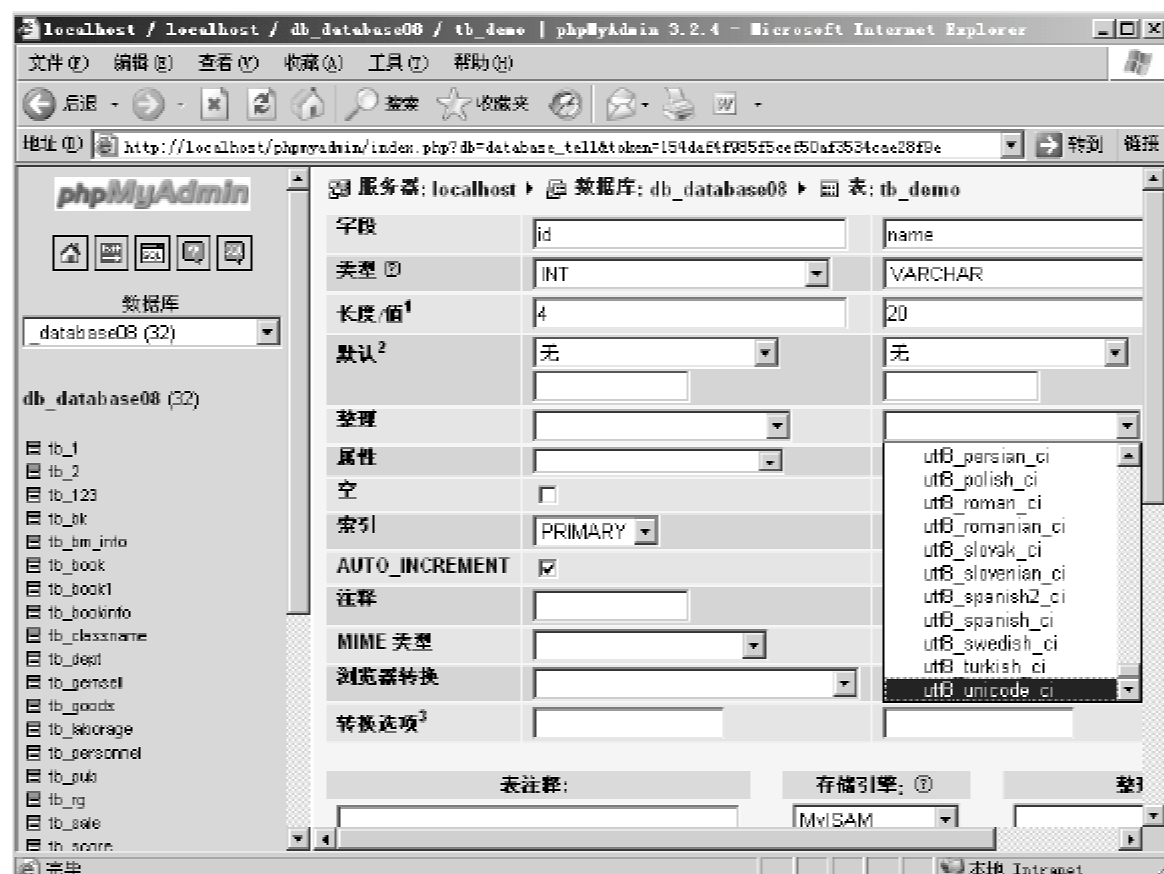


图 8.123　设置字段编码格式

秘笈心法

心法领悟 446：设置统一的编码格式。

在创建数据表字段时，一般将字符类型的数据字段设置为统一的编码格式，对于数值型数据字段，可以不进行编码格式的设置。

实例 447	phpMyAdmin 操作数据库 光盘位置：光盘\MR\08\447	高级 趣味指数：★★★★

实例说明

phpMyAdmin 可以实现在 cmd 命令行下对数据库的操作，而且操作更简单、直观。本实例通过 phpMyAdmin 实现对数据库的创建和删除操作。

关键技术

切记，在创建数据库时，一定要设置数据库的编码格式。注意，一旦执行删除操作，数据库将不能再恢复。

设计过程

（1）登录到 phpMyAdmin 主页，根据提示编写新建数据库的名称，并设置数据库的编码格式，如图 8.124 所示。

（2）单击“创建”按钮，创建数据库并自动选择创建的数据库为当前数据库。

（3）如果数据库已经没有任何作用，可以通过 phpMyAdmin 删除数据库。选择想要删除的数据库，单击导航栏中的“删除”按钮，根据提示决定是否删除，如图 8.125 所示。



图 8.124　创建数据库

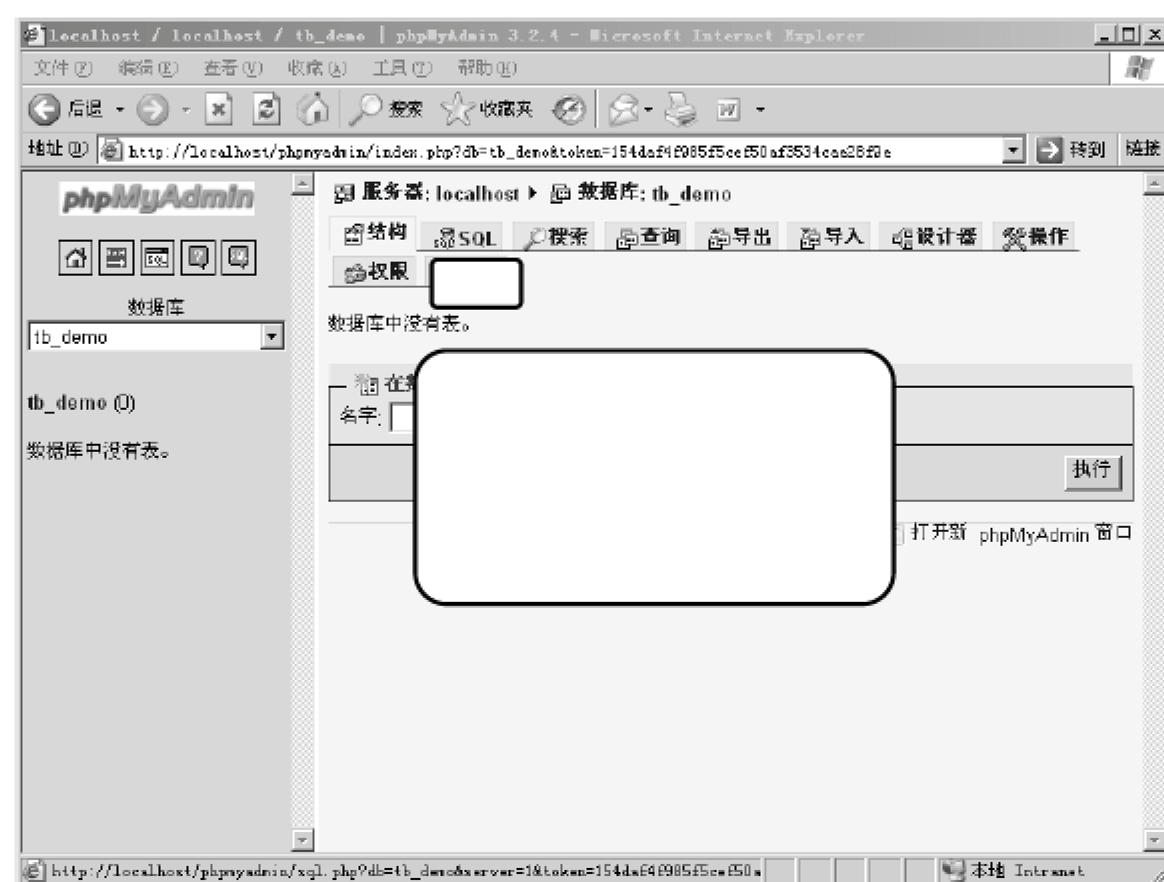


图 8.125　删除数据库

秘笈心法

心法领悟 447：phpMyAdmin 图形化工具与 cmd 命令下操作数据库。

通过 phpMyAdmin 图形化工具与在 cmd 命令下操作数据库没有任何区别，只是 phpMyAdmin 的设计者对 MySQL 数据库的相关指令进行了封装，使一些对 MySQL 数据库不甚了解的用户也可以操作数据库。

实例 448　phpMyAdmin 操作数据表　高级

光盘位置：光盘\MR\08\448　趣味指数：★★★★

实例说明

对数据表的操作一般包括创建和删除，本实例介绍应用 phpMyAdmin 图形化工具实现对数据表的创建和删除操作。

关键技术

想要创建数据表，首先要创建数据库，当数据库创建完成时自动选择已经创建好的数据库，根据提示编写数据表名称并创建数据表。而删除数据表时，需要选择想要删除的表，然后单击“删除”按钮即可完成。

设计过程

（1）登录到 phpMyAdmin 界面，创建数据库，选择数据库编码格式，单击“创建”按钮，根据提示编写数据表名称，并填写想要创建数据表的字段数目，如图 8.126 所示。

（2）进入编写数据库字段页，建立数据表结构，填写字段信息，其中包括字段名称、字段类型、字段长度、是否允许为空等，如图 8.127 所示。

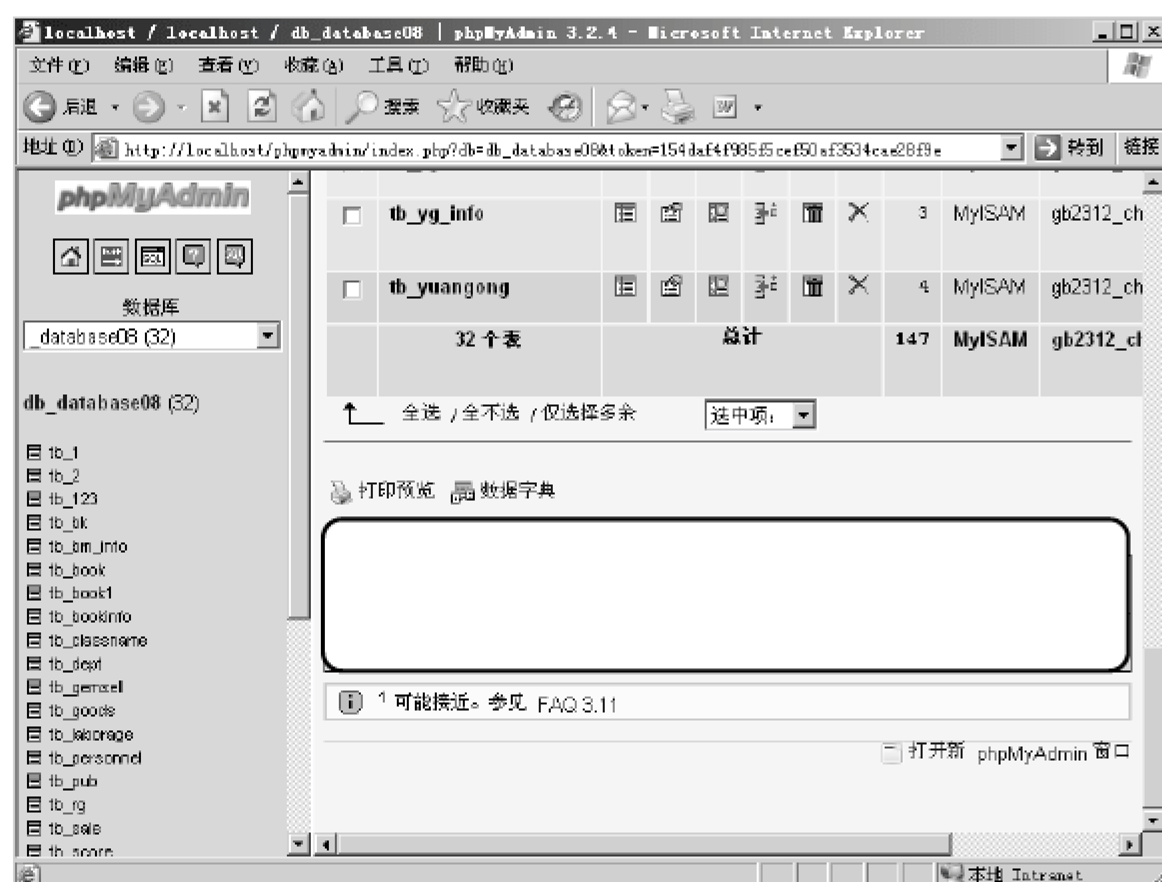


图 8.126　创建数据表

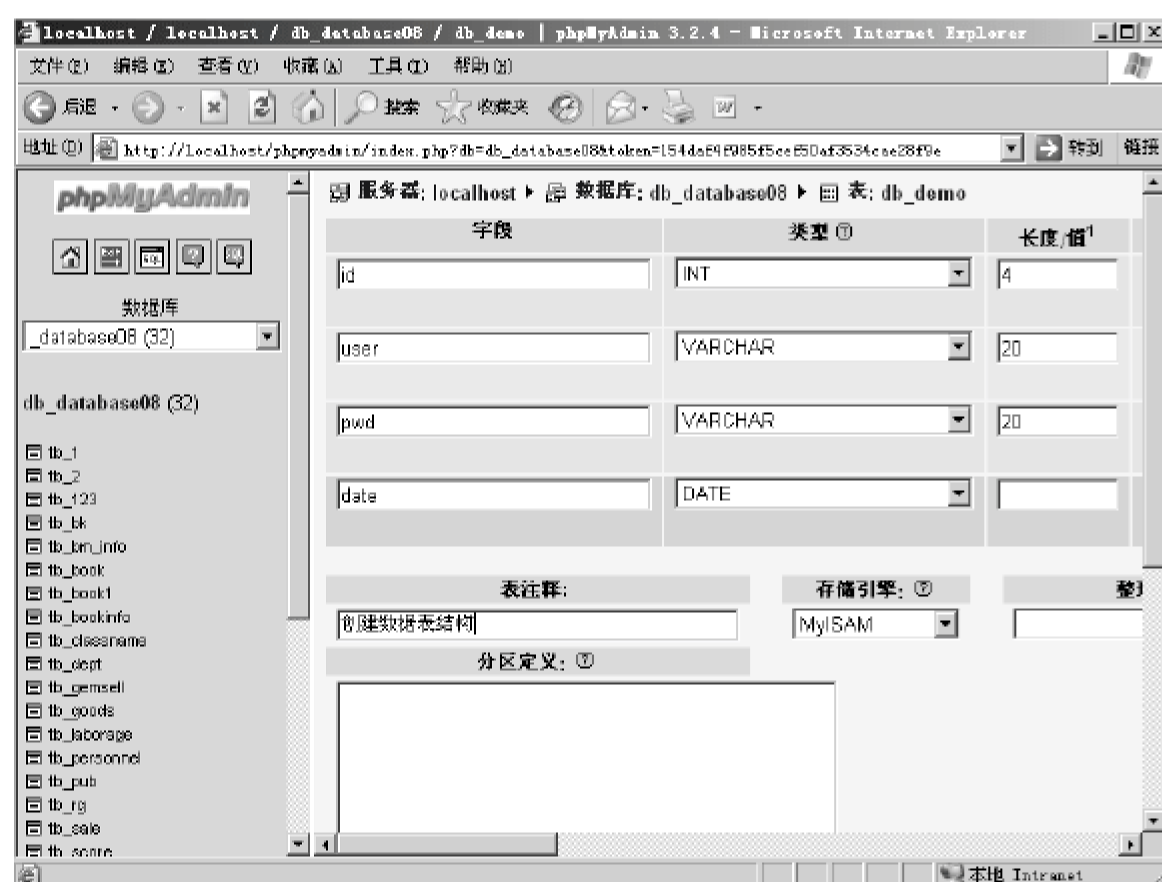


图 8.127　创建数据表结构

（3）单击“保存”按钮，完成数据表的创建。

（4）删除数据表时，首先选择要删除的数据表名称，并进入数据表操作界面，单击导航栏中的“删除”按钮，根据提示删除数据表，如图 8.128 所示。

秘笈心法

心法领悟 448：使用 phpMyAdmin 图形化工具创建数据表的注意事项。

在创建数据表结构时也要对数据表字段进行编码格式设置，而且必须为数据表指定主键、设置索引，否则会出现如图 8.129 所示的提示内容。

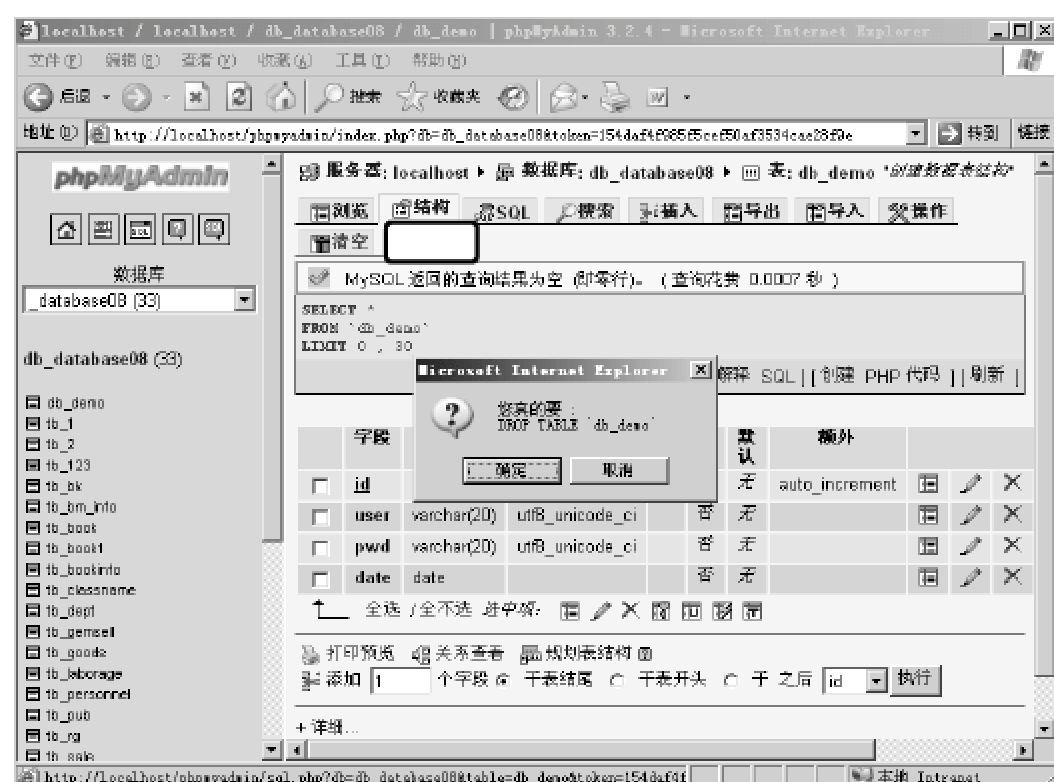


图 8.128　删除数据表

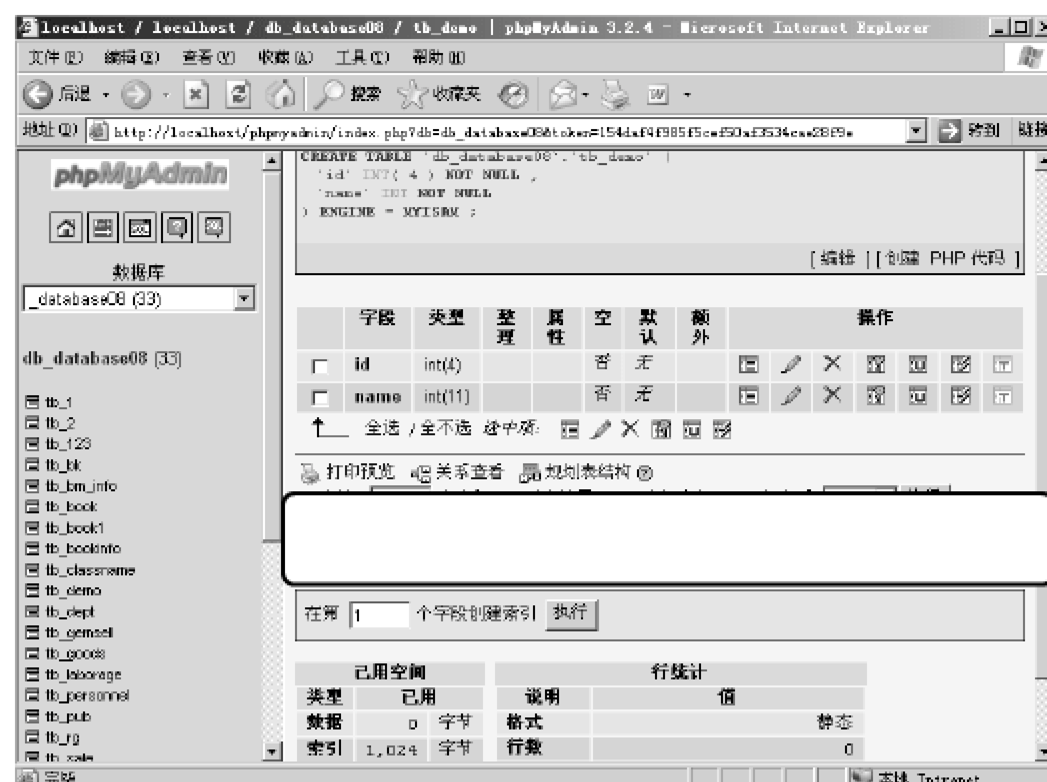


图 8.129　没有定义索引

实例 449　phpMyAdmin 操作数据

光盘位置：光盘\MR\08\449

高级

趣味指数：★★★★

实例说明

对 MySQL 数据库中数据的操作包括插入、删除、修改和查询等。本实例将介绍如何通过 phpMyAdmin 图形化管理工具完成上述的操作。

关键技术

phpMyAdmin 图形化管理工具操作数据的关键点如下：

☑　查询数据信息：进入表名为 tb_demo（以 tb_demo 为例）的数据表界面，单击导航栏中的“浏览”按钮。

☑　插入数据信息：单击导航栏中的“插入”按钮，根据提示插入信息。

☑　删除数据信息：单击删除字段的 ✕ 图标，删除一行数据。

☑　修改数据信息：单击修改字段的 ✎ 图标，修改一个或多个字段数据。

设计过程

（1）查询数据信息。首先进入 tb_demo 数据表，默认情况下直接显示数据表的信息。如果刚刚进行完其他操作，则单击导航栏中的“浏览”按钮，显示数据表信息，如图 8.130 所示。

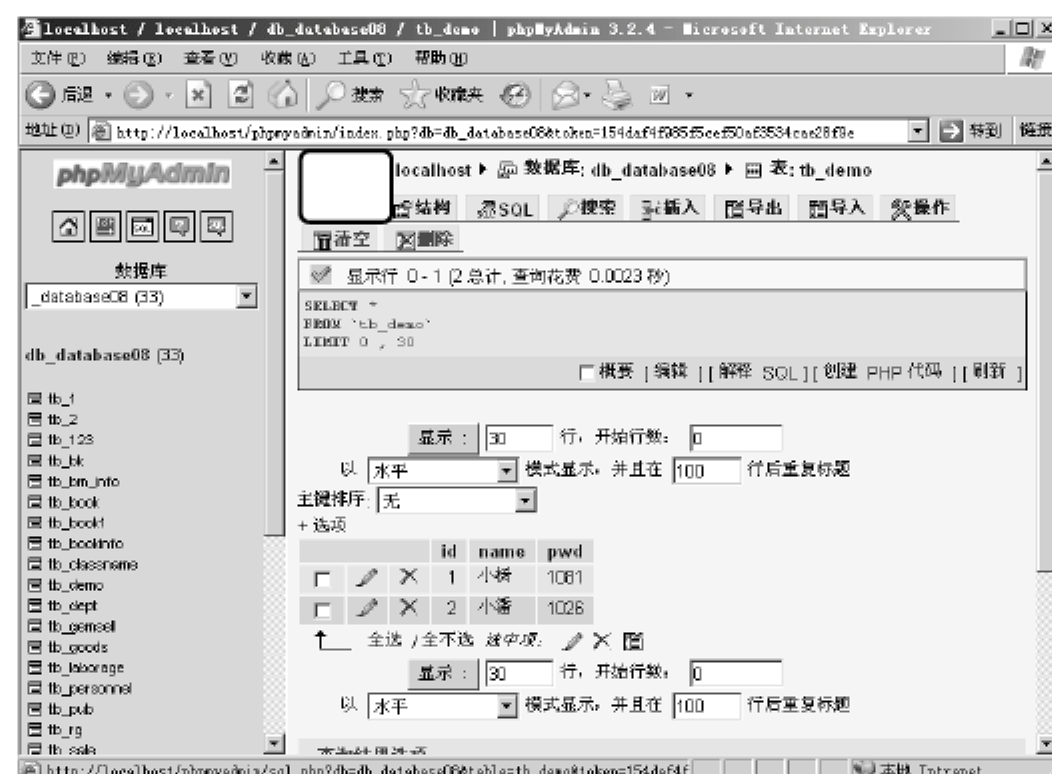


图 8.130　查询数据信息

（2）插入数据信息。单击导航栏中的“插入”按钮，进入插入信息界面，完成数据的添加，单击“执行”按钮，实现插入数据，如图 8.131 所示。

（3）删除数据信息。首先返回到数据浏览页，如果要删除单条数据信息，直接单击该条数据的删除按钮；如果要删除多条数据信息，可以选中数据信息前端的复选框，然后单击数据表信息下端的“选中项删除”按钮；如果要删除数据表中的所有信息，可以单击导航栏中的“清空”按钮，实现删除所有数据信息，如图 8.132 所示。

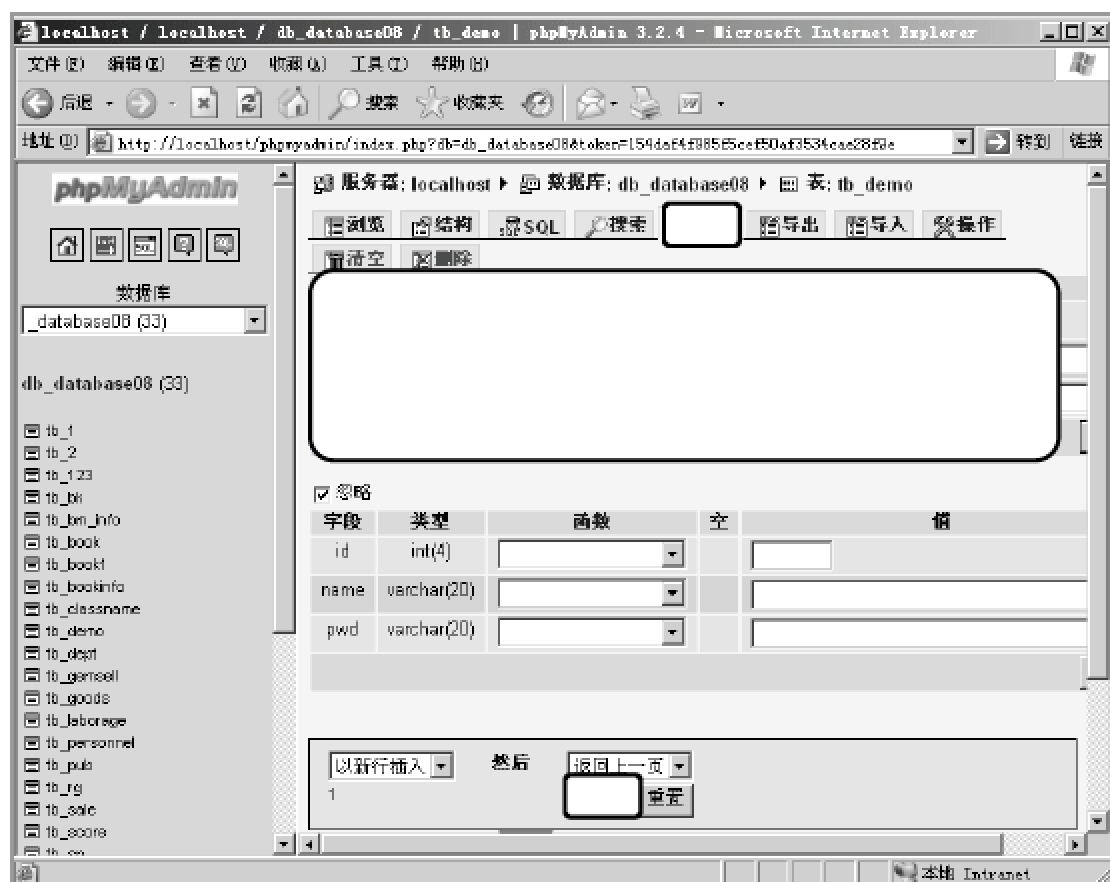


图 8.131　插入数据信息

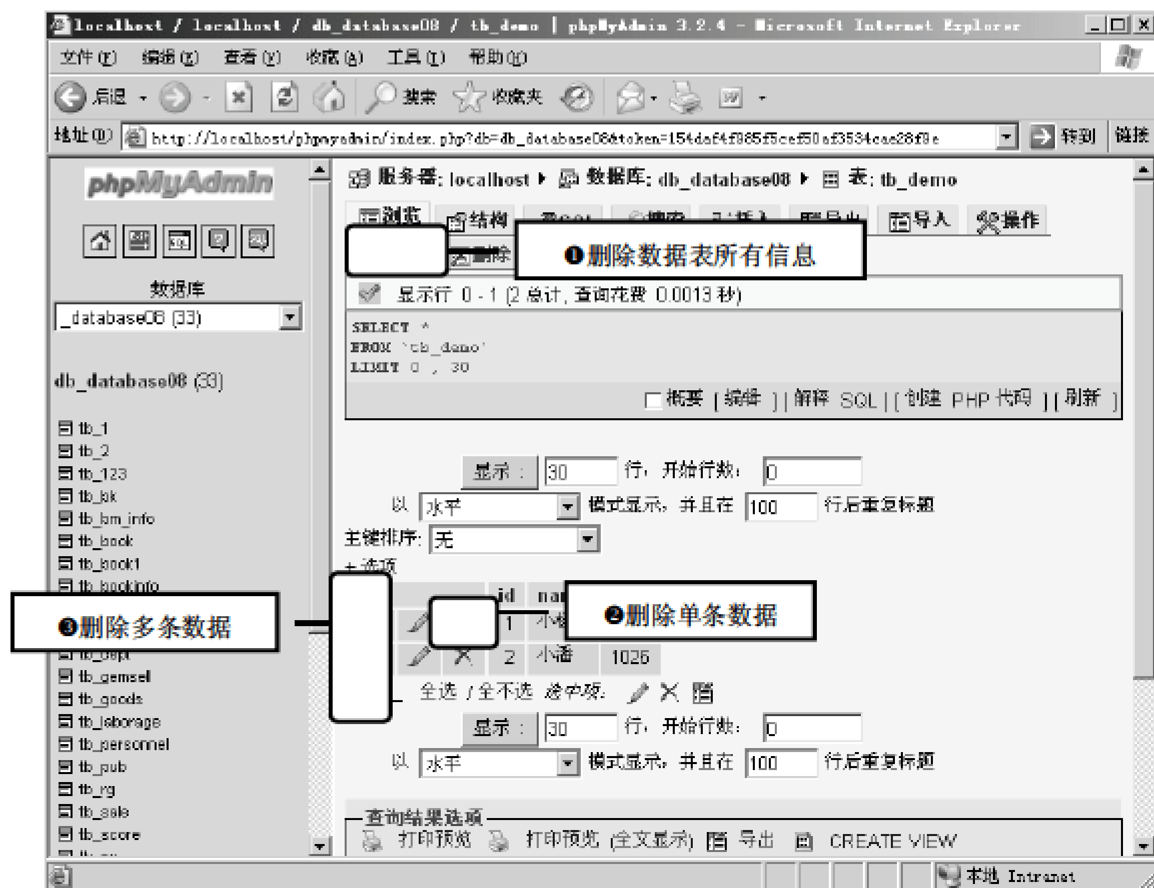


图 8.132　删除数据操作

（4）修改数据信息。修改数据信息与删除数据信息类似，可以分为修改单条数据信息和修改多条数据信息。修改单条数据信息时直接单击该条数据的修改按钮进入修改页面，即可实现修改；修改多条数据信息时，首先要勾选复选框，然后单击“选中项修改”按钮，进入修改页面，完成修改，如图 8.133 和图 8.134 所示。

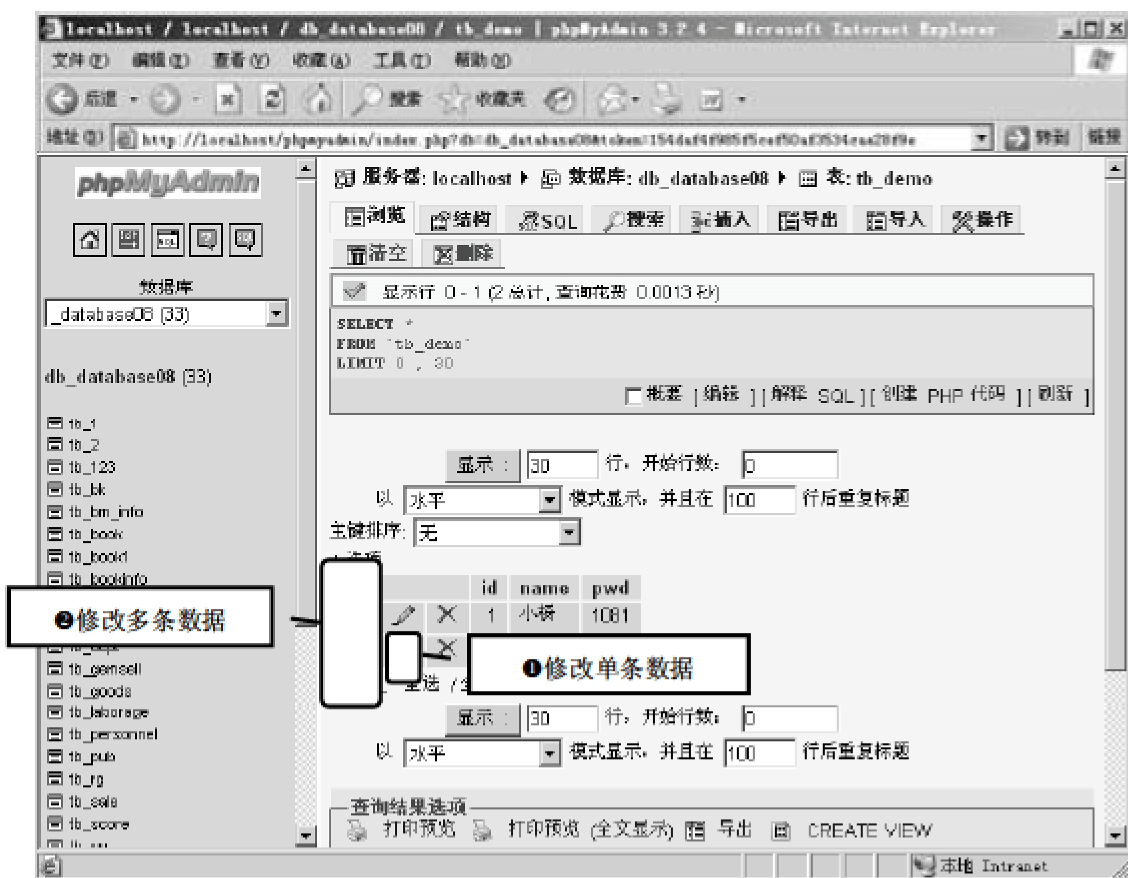


图 8.133　修改数据信息



图 8.134　更改数据信息页

秘笈心法

心法领悟 449：在 phpMyAdmin 图形化界面工具中修改操作与删除操作的区别。

如果在修改多条数据时，误选中不需更改的数据信息前的复选框，进入修改页面后可以不对此条数据做任何修改，此条数据信息不会有任何变化。但是如果在删除多条数据时，误选中不需删除的数据信息前的复选框，则单击“执行”按钮，信息会被删除。



9.1　PHP 的 MySQLi 扩展库

在 PHP 5.0 以后的版本中，为适应 MySQL 数据库的新特性重新编写了 MySQLi 扩展库。MySQLi 扩展库同时提供了面向过程和面向对象的两种调用接口，本节将通过具体实例讲解如何使用 MySQLi 扩展库对 MySQL 数据库进行操作。

实例 450　使用 MySQLi 扩展库连接 MySQL 数据库

光盘位置：光盘\MR\09\450

中级　趣味指数：★★★

实例说明

本实例主要实现使用 MySQLi 扩展库的面向对象的方式建立与 MySQL 数据库的连接。运行本实例，如图 9.1 所示，在页面中打印出数据库连接成功的提示信息，从而说明已经成功连接上 MySQL 数据库。



图 9.1　数据库连接成功

关键技术

使用 MySQLi 扩展库的面向对象的方法建立与 MySQL 数据库的连接，主要是使用 mysqli 类实现的，该类的构造方法的语法格式如下所示：

```
class mysqli{
    __construct ( [string host [, string username [, string passwd [, string dbname [, int port]]]]] )
}
```

mysqli 类的构造方法的参数说明，如表 9.1 所示。

表 9.1　类 mysqli 构造方法的参数说明

参数名称	说　明
host	MySQL 数据库服务器名称
username	MySQL 数据库服务器用户名
passwd	MySQL 数据库服务器密码
dbname	要连接的 MySQL 数据库的名称
port	MySQL 数据库服务器端口号，默认为 3306

设计过程

（1）创建数据库 db_database09。在 MySQL 中创建数据库可以通过 SQL 语句或 phpMyAdmin 图形化管理工具实现，本实例将采用命令方式创建 MySQL 数据库，其创建代码如下所示：

```
CREATE DATABASE `db_database09` ;
```

（2）使用函数 parse_ini_file()解析配置文件 config.ini 以获得数据库连接参数，然后使用 new 关键字对 mysqli

类进行实例化，最后使用函数 mysqli_connect_errno()来判断是否成功连接上了 MySQL 数据库，实现该过程的代码如下所示：

```
try{
    $config = parse_ini_file(realpath(dirname(__FILE__) . '/config/config.ini'));          //解析 config.ini 文件
    $mysqli = new mysqli($config['host'], $config['username'], $config['password'], $config['dbname']);   //对 mysqli 类进行实例化
    if(mysqli_connect_errno()){                                                            //判断是否成功连接上 MySQL 数据库
        throw new Exception('<br/>数据库连接错误！ ');                                      //如果连接错误，则抛出异常
    }else{
        echo '数据库连接成功！ ';                                                           //打印连接成功的提示
    }
}catch (Exception $e){                                                                     //捕获异常
    echo $e->getMessage();                                                                 //打印异常信息
}
```

注意：使用 MySQLi 扩展库完成对 MySQL 数据库的操作后，还应该通过调用 MySQLi 类实例对象的 close()方法释放资源。

秘笈心法

心法领悟 450：使用 MySQLi 扩展库的面向过程的方式连接 MySQL 数据库。

使用 MySQLi 扩展库不仅可以采用面向对象的方式建立与 MySQL 数据库的连接，为了适应不同开发习惯用户的需求，同时提供了面向过程的方式对 MySQL 数据库进行管理。使用 MySQLi 面向过程的方式建立与 MySQL 数据库的连接是通过 mysqli_connect()函数实现的，该函数的语法格式如下：

```
mysqli_connect( [string host [, string username [, string passwd [, string dbname [, int port]]]]] )
```

该函数参数的使用方式与 mysqli 类的构造方法相同，这里不再赘述。

实例 451 通过 MySQLi 扩展库实现数据库信息的检索

光盘位置：光盘\MR\09\451

初级

趣味指数：★★★

实例说明

在 Web 项目开发过程中，经常需要对大量数据信息进行查询操作，本实例主要介绍如何使用 MySQLi 扩展技术实现商品信息的检索。运行本实例，如图 9.2 所示，在图中文本框中输入要查找的商品名称，单击“查找”按钮，如果该商品存在则显示该商品的所有信息，否则提示没有查找到该商品。

商品信息管理系统

今天是 2010年7月28日 星期三

请输入商品名称：显示卡 查找

商品名称	商品价格	商品数量	进货日期
显示卡	500	34	2007-04-20 00:00:00

图 9.2 商品信息查询

关键技术

在 MySQLi 扩展库的面向对象的编程方式中，可以使用 mysqli 类中的 query()方法执行查询操作，该方法的语法格式如下所示：

```
mixed query ( string query [, int resultmode] )
```

参数说明：

query：要执行的查询语句。

resultmode：用于指定结果模式，这里可以取 MYSQLI_STORE_RESULT 和 MYSQLI_USE_RESULT 两个值，使用 MYSQLI_USE_RESULT 时在执行下一条查询前必须调用 mysqli_result 的 free()方法或 close()方法释放查询结果对象，否则查询失败。如果执行查询返回结果数据量较大一般使用 MYSQLI_USE_RESULT。

设计过程

（1）建立如图 9.2 所示的商品信息管理页面，在页面中建立按商品名称查询商品信息的表单，该页面的创建代码请详见本书附带光盘，同时建立 conn.php 文件实现与数据库的连接，该文件的代码如下：

```
$conn=new mysqli("localhost","root","111","db_database09");                    //建立与数据库的连接
$conn->query("set names utf8");                                                //设置数据库字符集
```

（2）当用户单击表单中的“查找”按钮后，将通过如下代码按照用户在表单中录入的关键字进行匹配查询。在下述代码中，首先判断用户是否提交了表单，如果是，则首先使用 include_once 语句包含 conn.php 文件，然后使用 mysqli 类实例后的对象调用 query()方法执行查询，并使用执行查询后的返回对象调用 fetch_array()方法获得查询结果集，如果结果集不为空则在页面中打印结果集，反之打印未查询到结果的提示信息。

```
<?php
if($_POST[submit]!="")                                                         //判断用户是否提交了表单
 {
    include_once("conn.php");                                                  //包含 conn.php 文件
    $name=$_POST[name];                                                        //获得提交关键字
    $sql=$conn->query("select * from tb_goods where name   like '%".$name."%'"); //执行查询
    $info=$sql->fetch_array(MYSQLI_ASSOC);                                     //获得查询结果集
    if($info==NULL){
 echo "<div align=center>没有查找到该商品!</div>";
    }else{
?>
<!--- 省略代码，请参见本书附带光盘   ->
      do{                                                                      //通过循环显现查询结果
  ?>
        <tr>
          <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[name];?></div></td>
          <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[price];?></div></td>
          <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[shuliang];?></div></td>
          <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[addtime];?></div></td>
        </tr>
  <?php
      }
    while($info=$sql->fetch_array(MYSQLI_ASSOC));
  ?>
    </table></td>
  </tr>
</table>
<?php
      }
}
?>
```

秘笈心法

心法领悟 451：获得 MySQLi 扩展库的结果集。

使用 mysqli 类中的 query()方法所返回的对象来调用 fetch_array()方法，可以获得查询结果集，该方法的语法格式如下：

```
fetch_array ( [int resulttype] )
```

fetch_array()方法中包含一个可省参数 resulttype，该参数用于指定返回结果的类型，其参数取值及说明如下：

MYSQLI_ASSOC：指定返回结果集为关联数组。

MYSQLI_NUM：指定结果集为索引数组。

MYSQLI_BOTH：指定结果集既包含关联数组又包含索引数组。



（3）分别执行 index.php 文件和 index_1.php 文件查看结果，如图 9.3 和图 9.4 所示。

秘笈心法

心法领悟 452：使用函数 mysqli_error()和 mysqli_errno()获得最后一次出错的错误信息和错误号。

本实例使用函数 mysqli_connect_error()和 mysqli_connect_errno()来获得连接数据库时的错误信息和错误号，那么如何获得 MySQLi 扩展库在操作 MySQL 数据库过程中的所有错误信息和错误号呢？这里可以使用函数 mysqli_error()和 mysqli_errno()实现，其中函数 mysqli_error()用于返回操作 MySQL 数据库时最后一次的错误信息，其语法格式如下：

```
string mysqli_error ( mysqli link )
```

参数 link 为函数 mysqli_connect()成功连接 MySQL 数据库的连接句柄。

mysqli_errno()用于返回操作 MySQL 数据库时最后一次的错误号，其语法格式如下：

```
int mysqli_errno ( mysqli link )
```

同样，参数 link 为函数 mysqli_connect()成功连接 MySQL 数据库的连接句柄。

实例 453　通过 MySQLi 扩展库实现多表查询

光盘位置：光盘\MR\09\453

初级

趣味指数：★★★

实例说明

一个成型的项目会包含很多数据表，并且表之间可能存在相互关联，本实例将讲解如何使用 MySQLi 扩展库执行多表之间的关联查询。运行本实例，如图 9.5 所示，在页面中打印出学生的学号、姓名和主要科目的成绩信息。

学生成绩管理系统

学号	姓名	语文成绩	外语成绩	数学成绩
0312315	刘小华	88	60	94
0312316	金星星	60	85	76
0312317	黄小全	56	90	75
0312318	李小林	76	86	78

图 9.5　学生成绩列表

关键技术

实现本实例的关键技术是如何通过主键实现多表之间的连接查询，下面是本实例实现多表查询的代码：

```
$sql=$conn->query("select * from tb_student, tb_score where tb_student.id=tb_score.sid");
```

实现多表之间的查询应在 from 子句后列出所有的表名，并且表名之间用逗号进行分割。同时应在 where 关键字中指明多表之间的连接条件，例如本实例中的 tb_student.id=tb_score.sid，表示学生表和成绩表之间通过学生表的 id 字段和成绩表的 sid 字段进行连接。

设计过程

（1）建立 conn.php 文件实现与 MySQL 数据库之间的连接，代码如下：

```
$conn=new mysqli("localhost","root","111","db_database09");      //建立与 MySQL 数据库的连接
$conn->query("set names utf8");                                  //设置字符编码
```

（2）建立 index.php 文件用于实现多表之间的查询，如果学生表和成绩表中有满足条件的记录则显示这些记录，否则提示没有相关信息，代码如下：

```
<?php
include_once("conn.php");                                         //包含 conn.php 文件
```

```
$sql=$conn->query("select * from tb_student,tb_score where tb_student.id=tb_score.sid");    //执行关联查询
$info=$sql->fetch_array(MYSQLI_ASSOC);                                                        //获得查询结果集
 if($info==NULL){                                                                             //未查到结果，将给出提示
       echo "暂无学生信息";
}else{
       do{
?>
<tr>
       <td height="20" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sno];?></div></td>
       <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[sname];?></div></td>
       <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[yw];?></div></td>
       <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[wy];?></div></td>
       <td bgcolor="#FFFFFF"><div align="center"><?php echo $info[sx];?></div></td>
</tr>
<?php
       }while(   $info=$sql->fetch_array(MYSQLI_ASSOC));                                    //通过循环打印成绩信息
}
?>
```

秘笈心法

心法领悟 453：使用隐式连接查询实现多表之间的关联查询。

实现本实例所采用的多表之间的关联查询称为显示关联查询，查询语句如下所示：

```
select * from tb_student,tb_score where tb_student.id=tb_score.sid
```

除了使用上述显示关联查询外，还可以使用内连接查询方式实现多表之间的隐式关联查询，如本实例的多表连接查询可以更改成如下形式：

```
select * from tb_student inner join tb_score on tb_student.id=tb_score.sid
```

上述内连接查询中，使用 inner join 子句指定要连接的表名，使用 on 子句指定两个表的连接条件，如果还有其他查询条件，可以指定 where 子句。

说明： 为了规范数据库查询技术，隐式查询方式将成为多表连接查询的主流方式。

实例 454 通过 MySQLi 扩展技术实现内存的回收

光盘位置：光盘\MR\09\454

初级

趣味指数：★★★

实例说明

为了有效利用服务器的内存空间，在获取较大结果集时，一旦结束操作，则有必要回收集合所需要的内存，由于 MySQLi 技术可以采用面向对象和面向过程两种编程方式，所以与之对应的内存回收方式也有两种，笔者将通过本实例讲解内存回收机制在项目开发过程中的意义和使用方法。运行本实例，如图 9.6 所示，图中为某班级学生主要科目成绩的列表，并在显示完该班成绩后对内存进行了回收。

学号	姓名	语文成绩	外语成绩	数学成绩
0312315	刘小华	88	60	94
0312316	金星星	60	85	76
0312317	黄小全	56	90	75
0312318	李小林	76	86	78

图 9.6　学生成绩列表

关键技术

PHP 为了迎合不同开发习惯的程序员，在 MySQLi 扩展技术中提供了面向对象和面向过程两种编程方式，所以实现内存回收也存在与之对应的两种方式。如果采用面向过程的编程方式可以通过如下函数实现内存回收：

```
bool mysqli_close ( object link)
```



法如下所示：

```
header('content-type:text/html; charset=UTF-8');
```

如果页面中包含规范的 HTML 代码，可以在 HTML 的 Meta 标记中设置页面编码，设置方法如下所示：

```
<meta http-equiv="content-type" content="text/html; charset=UTF-8" />
```

完成对 PHP 文件及页面编码的设置后，还需要设置数据库编码，一般需要进行两处设置。在创建数据表时要设置字段的字符编码，其设置方法如图 9.8 所示。



图 9.8　设置表中字段的字符编码

完成上述设置后，一般还需要在程序中采用编码方式设置数据库字符集，其设置方法可以通过 SQL 语句实现，例如本实例通过如下语句设置数据库字符集为 utf-8。

```
$conn->query("set names utf8");
```

注意：设置数据库字符编码为 utf-8 时，应该为 set names utf8 而非 set names utf-8，请不要多加字符“-”。

实例 455	用户登录 光盘位置：光盘\MR\09\455	初级 趣味指数：★★★

实例说明

在本实例中运用 mysqli 函数库中的函数，实现用户登录的功能，从中体会数据库连接函数（mysqli_connect()）和操作函数（mysqli_query()、mysqli_fetch_row）的应用。运行本实例，当输入正确的用户名和密码之后，将弹出如图 9.9 所示的页面。



图 9.9　用户登录

关键技术

在本实例中运用 mysqli_connect()函数连接数据库，运用 mysqli_connect_error()函数返回错误信息，运用 mysqli_fetch_row()函数获取查询结果的记录数，判断登录用户是否正确。

（1）mysqli_connect()函数，建立与 MySQL 数据库服务器的连接，其语法如下：

```
object mysqli_connect([string hostname [, string username [, string passwd [, string dbname [, int port [, string socket]]]]]])
```

mysqli_connect ()函数的参数说明如表 9.2 所示。

表 9.2　mysqli_connect()函数的参数说明

参　数	说　明
hostname	可选参数，MySQL 数据库服务器的地址
username	可选参数，MySQL 数据库的用户名
password	可选参数，MySQL 数据库的用户密码
dbname	可选参数，要连接的数据库名
port	可选参数，MySQL 数据库的端口号
socket	可选参数，用于指定管道端口

（2）mysqli_fetch_row()函数，根据所获取的记录生成数组，如果没有更多记录则返回 NULL，其语法如下：

```
mixed mysqli_fetch_row( mysqli_result result )
```

参数 result 为 mysqli_query()函数执行查询后的返回值。

设计过程

（1）建立 conn.php 文件，实现与 MySQL 数据库之间的连接。在 conn.php 文件中，首先使用 mysqli_connect()函数连接 db_database09 数据库，用户名是 root，密码为 111。然后，通过 mysqli_connect_error()函数定义错误信息。最后，通过 mysqli_query()函数设置数据库的编码格式，其代码如下：

```
<?php
$conn=mysqli_connect("localhost","root","111","db_database09");          //连接数据库
if (mysqli_connect_errno()) {
        printf("连接失败: %s\n", mysqli_connect_error());                 //返回错误信息
        exit();
}
mysqli_query($conn,"set names utf8");                                    //设置数据库编码格式
?>
```

（2）新建 index.php 文件，创建 form 表单，将用户登录的用户名和密码提交到 index_ok.php 文件处理。

（3）创建 index_ok.php 文件，获取表单提交的用户名和密码，根据 mysqli_fetch_row()函数的返回值判断登录的用户名和密码是否正确，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                     //设置文件编码格式
include("conn/conn.php");                                                //包含数据库连接文件
if($_POST['user']!="" || $_POST['pass']!=""){                            //判断用户名和密码是否正确
    $sql=mysqli_query($conn,"select * from tb_login where username='".$_POST['user']."' and password='".$_POST['pass']."'");
    $info=mysqli_fetch_row($sql);                                        //获取查询结果
    if($info==NULL){
        echo "<script>alert('登录失败！'); window.location.href='index.php'</script>";
    }else{
        echo "<script>alert('登录成功！'); window.location.href='index.php';</script>";
    }
}
?>
```

秘笈心法

心法领悟 455：mysqli_fetch_row()函数的应用。

根据 mysqli_fetch_row()函数的返回值不但可以判断是否存在查询结果，如果与 while 语句结合使用，还可以输出查询结果中的数据。

实例 456	用户注册 光盘位置：光盘\MR\09\456	初级 趣味指数：★★★

实例说明

在本实例中运用 mysqli 函数库中的 mysqli_real_connect()函数实现与 MySQL 数据库的连接，从而实现用户注册的功能，并且介绍一种检测 SQL 语句中错误的方法。运行本实例，输入用户的注册信息，如果用户输入的用户名已经存在，则输出用户名已经存在的提示信息，否则提示用户注册成功，其运行结果如图 9.10 所示。

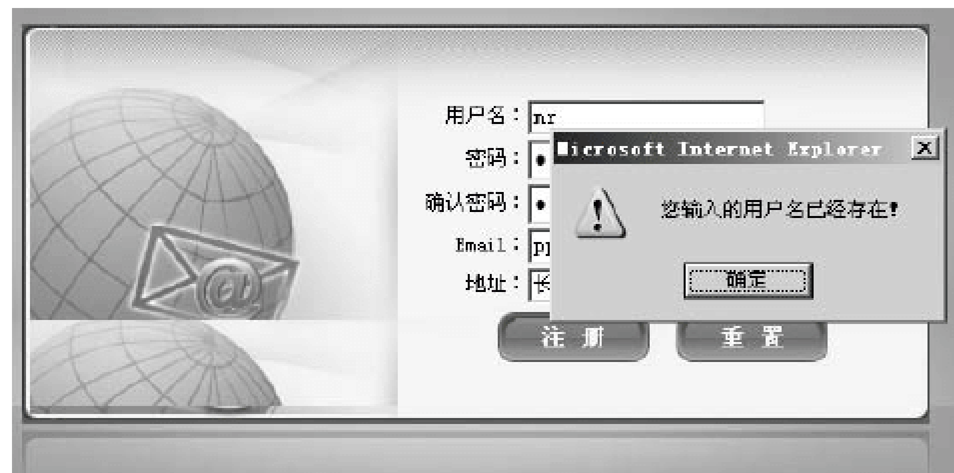


图 9.10　用户注册



```
}
mysqli_query($conn,"set names gb2312");                              //设置数据库编码格式
?>
```

（2）新建 index.php 文件，创建 form 表单，将用户注册信息提交到 index_ok.php 文件进行处理。

（3）创建 index_ok.php 文件，获取表单提交的用户注册信息，首先，判断提交的用户名是否被占用，如果未被占用，则将用户注册信息添加到 tb_reg 表中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=gb2312" );                //设置页面编码
include("conn/conn.php");                                            //包含数据库连接文件
if($_POST['user']!="" || $_POST['pwd1']!="" ){                       //判断输入的用户名和密码是否为空
        $check="select name from tb_reg where name='".$_POST['user']."'";  //定义验证用户名是否被占用的 SQL
        $result=mysqli_query($conn,$check);                          //执行查询语句
$info=mysqli_fetch_row($result);                                     //获取查询结果
      if($info!=NULL){
        echo "<script>alert('您输入的用户名已经存在！'); window.location.href='index.php';</script>";
      }else{
        $sql="insert into tb_reg(name,pwd,email,address)
values('".$_POST['user']."','".md5($_POST['pwd1'])."','".$_POST['email']."','".$_POST['address']."')";
        $result=mysqli_query($conn,$sql);
        echo mysqli_error($conn);                                    //返回 SQL 语句中的错误信息
        if($result){
                echo "<script>alert('注册成功!'); window.location.href='index.php';</script>";
        }else{
                        echo "<script>alert('注册失败!'); window.location.href='index.php';</script>";
        }
}
}
?>
```

秘笈心法

心法领悟 456：检测 SQL 语句中的错误。

在 mysqli 函数库中，可以通过 mysqli_error()函数来检测程序 SQL 语句中的错误，以帮助我们及时地发现 SQL 语句中的问题。

实例 457	数据的浏览 光盘位置：光盘\MR\09\457	初级 趣味指数：★★★

实例说明

在前面的实例中介绍了如何通过 mysqli 函数库中的函数实现用户注册和登录的功能，在本实例中将介绍如何通过 mysqli 函数库中的函数完成数据的浏览操作。运行本实例，将循环输出数据库中指定数据表的数据，其运行结果如图 9.11 所示。

图 9.11　浏览注册用户

关键技术

在本实例中，获取查询结果中数据应用的是 mysqli_fetch_object()函数，并结合 while 语句完成其返回对象中数据的输出。

mysqli_fetch_object()函数，返回一个由执行查询后生成的表字段组成的对象，其语法如下：

```
mixed mysqli_fetch_object(object result)
```

参数 result 为 mysqli_query()函数执行查询的返回值。

获取返回对象中数据的格式如下：

```
<?php echo $obj->name; ?>
```

其中$obj 为返回的对象，name 为字段名称。

设计过程

（1）建立 conn.php 文件实现与 MySQL 数据库之间的连接。

（2）新建 index.php 文件，创建 form 表单，将用户登录信息提交到 index_ok.php 文件进行处理。

（3）创建 index_ok.php 文件，获取表单提交的用户登录信息，判断提交的用户名和密码是否正确，如果正确，则将登录用户名存储到 SESSION 变量中，并跳转到 main.php 文件，其代码如下：

```
<?php
session_start();                                                    //初始化 session 变量
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include("conn/conn.php");                                           //包含数据库连接文件
if($_POST['user']!="" || $_POST['pass']!=""){
      $sql=mysqli_query($conn,"select * from tb_login where username='".$_POST['user']."' and password='".$_POST['pass']."'");
      $info=mysqli_fetch_row($sql);
      if($info==NULL){
         echo "<script>alert('登录失败！ '); window.location.href='index.php'</script>";
      }else{
 $_SESSION['user']=$_POST['user'];
         echo "<script>alert('登录成功！ '); window.location.href='main.php';</script>";
      }
}
?>
```

（4）创建 main.php 文件，连接数据库类，执行查询语句，应用 mysqli_fetch_object()函数和 while 语句完成数据库中数据的循环输出，其关键代码如下：

```
<?php
session_start();
include("conn/conn.php");
if($_SESSION['user']==true){
?>
//省略部分代码
<?php
         $sql=mysqli_query($conn,"select name,email,address from tb_reg ");
 while($obj=mysqli_fetch_object($sql)){
 ?>
<tr>
            <td height="22" align="center"><?php echo $obj->name;?></td>
             <td align="center"><?php echo substr($obj->email,0,20);?></td>
             <td align="center"><?php echo $obj->address;?></td>
      </tr>
<?php
          }
?>
//省略部分代码
<?php
}else{
         echo "<script>alert('您不具备访问权限！ ');window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 457：返回查询结果集中的数据。

返回查询结果集中的数据，不但可以使用 mysqli_fetch_object()函数，还可以使用 mysqli_fetch_array()、mysqli_fetch_assoc()和 mysqli_fetch_row()函数。

通过 mysqli_fetch_array()函数返回的查询结果集可以使用数字索引或者关联索引来读取数据；而通过 mysqli_fetch_assoc()函数返回的查询结果集必须使用关联索引来获取数据，通过 mysqli_fetch_row()函数返回的查询结果集必须使用数字索引来获取数据。

实例 458　数据库、数据表中数据的动态输出

光盘位置：光盘\MR\09\458

初级

趣味指数：★★★

实例说明

在本实例中应用 mysqli 函数库中的函数动态地对数据库和数据表进行操作，统计数据表中的字段和记录数，并且循环输出数据表中的数据，其运行结果如图 9.12 所示。



图 9.12　浏览注册用户

关键技术

（1）统计查询结果中的字段数应用的是 mysqli_num_fields()函数，其语法如下：

```
int mysqli_num_fields(mysqli_result result)
```

参数 result 为 mysqli_query()、mysqli_store_result()或 mysqli_use_result()函数执行查询后的返回值。

（2）统计查询结果中的记录数应用的是 mysqli_num_rows 函数，其语法如下：

```
int mysqli_num_rows(mysqli_result result)
```

参数 result 为 mysqli_query()、mysqli_store_result()或 mysqli_use_result()函数执行查询后的返回值。

（3）获取数据表中字段信息应用的是 mysqli_fetch_fields()函数。它返回一个对象数组，该数组包含某表字段的信息，如果该表的字段为空，则返回 FALSE，其语法如下：

```
array mysqli_fetch_fields(mysqli_result result)
```

参数 result 为 mysqli_query()函数的返回值，返回的对象数组可以调用的属性如表 9.5 所示。

表 9.5　mysqli_fetch_fields()函数返回的对象数组可以调用的属性说明

属　　性	说　　明
name	返回字段的名称
orgname	返回字段的原始名称
table	返回字段所属的表名
orgtable	返回字段所属的原始表名
def	返回该字段的默认值
max_length	返回字段的最大宽度
length	字段的宽度
charsetnr	返回字段所属的字符集
flags	返回一个描述字段的位标记
type	返回字段的数据类型
decimals	返回字段小数部分的位数

在本实例中只是应用 name 属性，获取数据表字段的名称。

设计过程

在本实例中，创建 index.php 文件。首先，创建 form 表单，将要访问的数据库和数据表名称提交到本页。然后，获取提交的数据库、数据表名称，执行连接操作，连接成功后，统计数据表中字段和记录的总数。最后，循环输出指定数据表中的数据，其关键代码如下：

```
<?php
if($_POST[submit]!=""){
 $dbname=$_POST[dbname];
 $tbname=$_POST[tbname];
 $conn=mysqli_connect("localhost","root","111",$dbname);
 if(mysqli_connect_errno()) {
         printf("数据库连接失败: %s\n", mysqli_connect_error());
            exit();
 }
 mysqli_query($conn,"set names utf8");
 $result=mysqli_query($conn,"select * from ".$tbname."");
 echo "该表共有字段 ".mysqli_num_fields($result)." 个!";
 echo "该表共有记录 ".mysqli_num_rows($result)." 条!";
}
?>
                </td>
         </tr>
      </table>
<?php
if($conn){
?>
<table width="550" bordercolor="#FFFFFF" bgcolor="#FFCC33">
 <tr>
<?php
 for($i=0; $i<mysqli_num_fields($result);$i++){
         $obj=mysqli_fetch_field($result);
?>
         <td height="25" bgcolor="#FFFFFF" ><div align="center"><?php echo $obj->name;?></div></td>
<?php
 }
?>
 </tr>
<?php
 $query = "select * from ".$tbname."";
 $sql=mysqli_query($conn,$query);
 $info=mysqli_fetch_array($sql);
 if($info==NULL){
         echo "暂无员工信息!";
 }else{
         do{
?>
 <tr>
         <?php
                 for($i=0; $i<mysqli_num_fields($sql);$i++){
         ?>
                         <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[$i];?></div></td>
         <?php
                 }
         ?>
 </tr>
<?php
         }while($info=mysqli_fetch_array($sql));
 }
 mysqli_close($conn);
?>
            </table>
<?php
}
?>
```



设计过程

（1）建立数据库及数据库表。本实例中将数据表命名为 tb_admin，其创建过程可以通过 phpMyAdmin 实现。

（2）完成数据表的创建后，在命令提示符下建立视图，创建视图的代码如下：

```
create view chkadmin as select name, pwd from tb_admin
```

创建的视图 chkadmin 中只包含 name 和 pwd 两个字段信息，这样使得程序的其他字段信息不可见，从而可以有效提高程序的安全性。

（3）采用 PHP 中的 MySQL 函数库建立与 MySQL 数据库的连接。首先使用 mysql_connect()函数连接 MySQL 数据库，然后使用函数 mysql_select_db()选择数据库 db_database09，最后使用函数 mysql_query()执行 set names utf8 语句来设置数据库的编码为 utf-8 编码，实现该过程的代码如下：

```
$conn=mysql_connect("localhost","root","111");          //建立与 MySQL 数据库的连接
mysql_select_db("db_database09",$conn);                 //选择数据库
mysql_query("set names utf8");                          //设置数据库字符编码
```

（4）判断用户是否单击了“进入”按钮，如果是，则通过视图 chkadmin 对用户身份进行验证，该过程的代码如下：

```
if($_POST[submit]!=""){
    include_once("conn.php");                           //包含 conn.php 文件
    $name=$_POST[name];                                 //接收提交的用户名
    $pwd=$_POST[pwd];                                   //接收提交的用户密码
    $sql=mysql_query("select * from chkadmin where name='".$name."' and
pwd='".$pwd."'",$conn);                                 //执行查询
    $info=mysql_fetch_array($sql);                      //返回结果集
    if($info==false){                                   //判断用户名和密码是否正确
        echo "<script>alert('用户名或密码输入错误!');history.back();</script>";
        exit;
    }else{
        echo "<br><div align=center>登录成功!</div>"; //如果正确则提示登录成功
    }
}
```

秘笈心法

心法领悟 459：使用视图的实际意义。

视图在项目开发过程中具有非常重要的实际意义，主要表现在如下两点：

☑ 使用视图可以使查询更加简化。在程序编写中，如果所要执行的查询较为复杂，可以将其转化为视图，从而简化查询。

☑ 使用视图还可以提高程序的安全性。如果只想让用户访问某表中的某几个指定的字段，可以创建一个只包含这些字段的视图，而不用将整个表的所有字段的访问权限都分配给该用户。

技巧：虽然不使用视图也可以完成对数据库的操作，但在字段数量较多或安全性要求较高的模块中建议通过视图完成查询，从而有效地提高程序的可维护性和安全性。

实例 460	修改 MySQL 数据库中的视图 光盘位置：光盘\MR\09\460	初级 趣味指数：★★★

实例说明

通过本实例主要讲解如何对已经创建完成的 MySQL 视图进行修改，在制作本实例时，首先需要在数据库中创建数据表 tb_changeview，其结构如图 9.15 所示，然后创建视图 userinfo，该视图中包含 tb_changeview 表中所有字段，完成上述操作后，使用 alter view 语句对视图进行修改。

	字段	类型	整理	属性	Null	默认	额外	操作
☐	id	int(8)			否		auto_increment	
☐	username	varchar(50)	gb2312_chinese_ci		否			
☐	userpwd	varchar(50)	gb2312_chinese_ci		否			

图 9.15　创建表 changeview

关键技术

在 MySQL 中修改视图可以通过 alter view 语句实现，alter view 语句的具体使用说明如下：

```
alter view [algorithm={merge | temptable | undefined} ]view view_name [(column_list)] as select_statement[with [cascaded | local] check option]
```

参数说明：

algorithm：该参数已经在创建视图中作了介绍，这里不再赘述。

view_name：视图的名称。

select_statement：SQL 语句，用于限定视图。

设计过程

（1）使用 phpMyAdmin 创建数据库及数据库表，本实例中将数据表命名为 tb_changeview。

（2）使用 create view 语句创建视图 userinfo，其创建代码如下所示：

```
create view userinfo as select * from tb_changeview
```

（3）使用 select 语句查询视图 userinfo，查询代码如下所示：

```
select * from userinfo
```

查看执行结果，可以发现 tb_changeview 表中的所有字段都一一列出，如图 9.16 所示。

（4）下面开始修改视图，要求修改后只列出 username 字段，代码如下：

```
alter view userinfo as select * from tb_changeview
```

（5）重新查询视图 userinfo，从结果中可以发现，此时只列出 username 一个字段，从而说明视图修改成功，如图 9.17 所示。

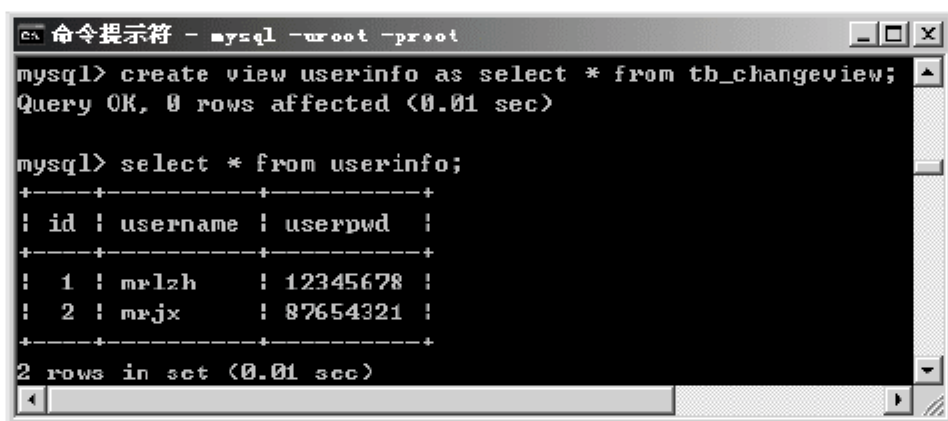


图 9.16　创建新视图

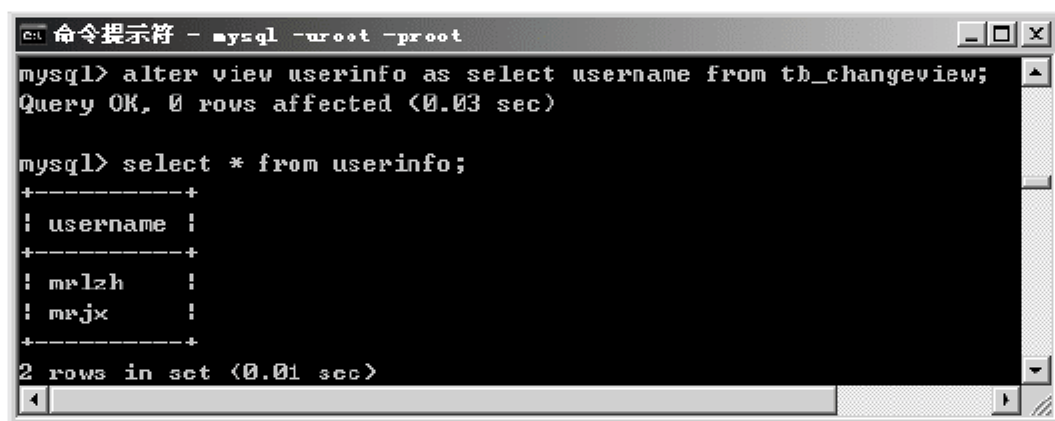


图 9.17　修改视图

注意：如果数据库采用 utf-8 编码，在命令提示符下通过 select 语句显示查询结果时会出现中文乱码现象。

秘笈心法

心法领悟 460：在 phpMyAdmin 的查询窗口中创建和修改视图。

在实现本实例时，创建视图的过程是在 Windows 系统的命令提示符窗口中实现的，为了提高开发效率、简化程序开发人员的操作，MySQL 数据库的创建和维护过程一般在 MySQL 的图形管理工具 phpMyAdmin 中进行，phpMyAdmin 为 MySQL 数据库提供了较为完善的创建、维护及备份等操作。同样，如果使用 phpMyAdmin 的查询窗口也可以执行 SQL 命令，所以可以在该查询窗口中创建视图，其具体创建过程如下：

（1）打开 phpMyAdmin 管理页面，并选择要管理的数据库，单击如图 9.18 所示的“查询窗口”按钮来打开 phpMyAdmin 的查询窗口。

（2）phpMyAdmin 的查询窗口如图 9.19 所示，可以在该窗口中输入 SQL 命令来实现对指定数据库及表的查询操作。例如在该查询窗口中输入创建视图的 SQL 命令，然后单击图中的“执行”按钮即可成功地实现视图的创建操作。



秘笈心法

心法领悟 461：在 phpMyAdmin 的命令窗口中删除视图。

要删除指定的视图，可以在 phpMyAdmin 的命令窗口中实现。首先打开 phpMyAdmin 的命令窗口，如图 9.22 所示，然后在该窗口中输入 drop view 语句，最后单击“执行”按钮即可成功删除指定的视图。

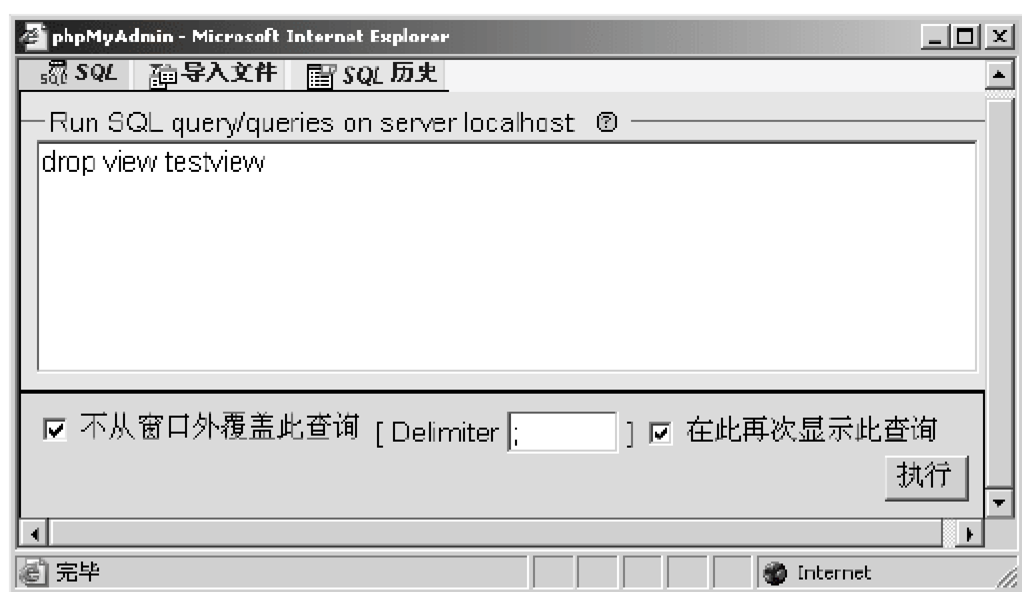


图 9.22　在 phpMyAdmin 的命令窗口中删除视图

实例 462　使用 MySQL 视图查询学生成绩信息

光盘位置：光盘\MR\09\462

初级

趣味指数：★★★

实例说明

在实际项目开发过程中数据表中可能有很多字段，但某个模块可能只需要其中的几个。为了提高查询速度和简化操作，可以将该模块需要的字段单独提取出来放在某个视图中，例如本实例涉及到学生表和成绩表，在建立的视图中只含有与学生成绩有关的字段，如图 9.23 所示。

运行本实例，如图 9.24 所示，图中查询结果显示的内容即为视图中所有字段中的内容。

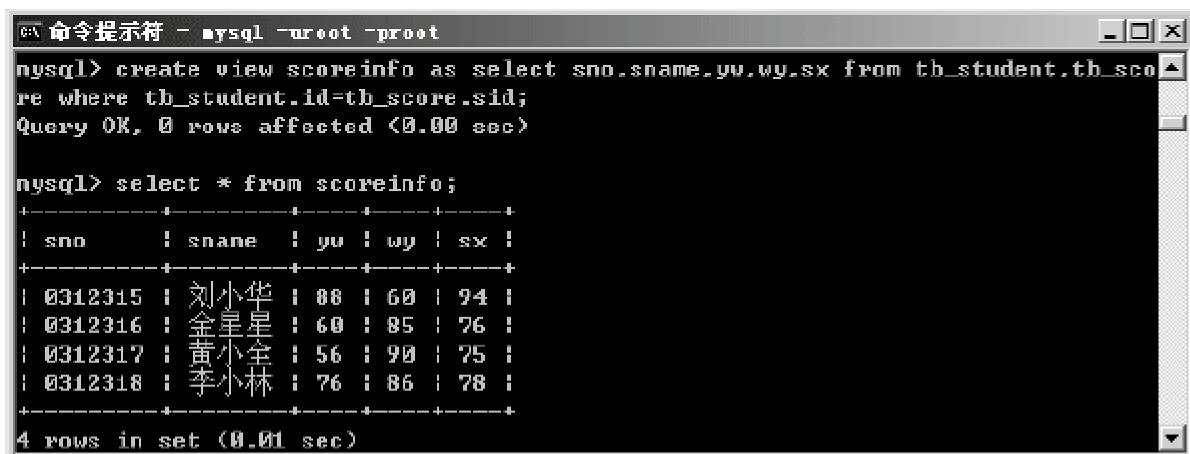


图 9.23　创建视图

视图的应用　学生成绩列表

学号	姓名	语文成绩	外语成绩	数学成绩
0312315	刘小华	88	60	94
0312316	金星星	60	85	76
0312317	黄小全	56	90	75
0312318	李小林	76	86	78

图 9.24　学生成绩列表

关键技术

本实例创建的视图涉及到多表查询，这说明多个表之间可以通过视图来组合为一个整体，这样对视图的操作就相当于多表查询。但如果这些表中有相同的字段，必须按如下方式书写重名的字段：

```
表名 1.字段名,表名 2.字段名……
```

设计过程

（1）创建视图 scoreinfo，通过该视图显示学生成绩信息，该视图的创建代码如下所示：

```
create view scoreinfo as select sno,sname,yw,wy,sx from tb_student,tb_score where tb_student.id=tb_score.sid
```

（2）建立数据库连接文件 conn.php，实现与 MySQL 数据库的连接，并设置数据库字符集为 utf-8 编码，代码如下：



```
else
set variable='PHP';
end if;
insert into tb (name) values (variable);
end;
```

在 MySQL 中，存储过程的建立以关键字 create procedure 开始，后面紧跟存储过程的名称和参数。MySQL 的存储过程名称不区分大小写，例如 PROCE1()和 proce1()代表同一存储过程名。存储过程名不能与 MySQL 数据库中的内建函数重名。

存储过程的参数一般由 3 部分组成。第 1 部分可以是 in、out 或 inout。in 表示向存储过程中传入参数；out 表示向外传出参数；inout 表示定义的参数可传入存储过程并可以被存储过程修改后传出，存储过程默认为传入参数，所以参数 in 可以省略。第 2 部分为参数名。第 3 部分为参数的类型，该类型为 MySQL 数据库中所有可用的字段类型，如果有多个参数，参数之间可以用逗号进行分隔。

MySQL 存储过程的语句块以 begin 开始，以 end 结束。语句体中可以包含变量的声明、控制语句、SQL 查询语句等。由于存储过程内部语句要以分号结束，所以在定义存储过程前应将语句结束标志“;”更改为其他字符，并且该字符应该在存储过程中出现的几率较低，可以用关键字 delimiter 转换，例如：

```
mysql>delimiter //
```

存储过程创建之后，可用如下语句进行删除：

```
drop procedure proc_name
```

参数 proc_name 指存储过程名。

设计过程

MySQL 存储过程是在命令提示符下创建的，所以首先应该打开“命令提示符”窗口。

（1）进入“命令提示符”窗口后，首先应该登录 MySQL 数据库服务器，在命令提示符下输入如下命令：

```
mysql –u 用户名 –p 用户密码
```

（2）更改语句结束符号。本实例将语句结束符更改为“//”，代码如下：

```
delimiter //
```

（3）创建存储过程前应首先选择某个数据库，代码如下：

```
use 数据库名
```

（4）创建存储过程。

（5）通过 call 语句调用存储过程。

秘笈心法

心法领悟 463：使用存储过程的优点。

☑ 使用存储过程进行项目开发可以提高程序的可移植性。在存储过程中，可以编写事务处理逻辑，并可以被其他程序所重用。

☑ 存储过程可以节省大量的工作，不需要重复执行同样的逻辑运算。

实例 464	使用存储过程实现用户登录 光盘位置：光盘\MR\09\464	初级 趣味指数：★★★

实例说明

通过本实例主要讲解如何使用 MySQL 存储过程实现用户登录身份的验证。运行本实例，如图 9.25 所示，在图中登录窗口的文本框中输入用户名和密码，然后单击“登录”按钮，如果用户录入的用户名和密码正确，则会将页面定向到如图 9.26 所示的登录成功提示页。



图 9.25　用户登录窗口



图 9.26　登录成功的提示信息

关键技术

在创建本实例的存储过程中，为了防止结束符冲突，首先使用 delimiter 语句将 MySQL 默认的结束符“;”更改为“//”，然后创建存储过程，其创建代码如图 9.27 所示。

```
C:\>mysql -uroot -proot
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14
Server version: 5.0.51b-community-nt-log MySQL Community Edition (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use db_database09;
Database changed
mysql> delimiter //
mysql> create procedure pro_login(in un varchar(50), in pwd varchar(50))
    -> begin
    -> select * from tb_login where username=un and password = pwd;
    -> end;
    -> //
Query OK, 0 rows affected (0.06 sec)
```

图 9.27　创建用户登录验证的存储过程

创建完成上述存储过程后，在程序中使用 call 语句调用要执行的存储过程，本实例使用 MySQLi 扩展库执行存储过程，其实现代码如下所示：

```
$sql = $mysqli->query("call pro_login('".$username."', '".$password."')");
```

设计过程

（1）通过 MySQLi 扩展库建立与 MySQL 数据库的连接，并设置数据库字符集为 utf-8，实现代码如下所示：

```
$mysqli=new mysqli("localhost","root","111","db_database09");          //建立与 MySQL 数据库的连接
$mysqli->query("set names utf8");                                        //设置数据库字符编码为 utf-8
```

（2）建立用户登录表单，当用户在表单中录入用户名和密码，并单击“提交”按钮后，通过如下代码验证用户的登录信息是否正确：

```
if(isset($_POST['username']) && trim($_POST['username'])!='')                //判断用户是否提交了表单
{
    require_once 'Db.php';                                                   //包含 Db.php 文件
    $username = trim($_POST['username']);                                    //获得提交的用户名
    $password = trim($_POST['password']);                                    //获得提交的用户密码
    $sql = $mysqli->query("call pro_login('".$username."', '".$password."')");  //调用存储过程，执行查询
    $info = $sql->fetch_array(MYSQLI_ASSOC);                                 //获得查询第一条记录
    if($info != null){                                                       //判读是否为 null 值
        $_SESSION['loginUsername'] =   $username;                            //将用户名保存到 session 中
        echo '<script>window.location.href="success.php";</script>';         //重定向到 success.php 页面
    }else {
        echo '<div style="width:300px; height:30px; line-height:30px; border:1px solid #E59B04; background-color:#FCF2E0; color:#FF0000;">用户
```



设计过程

（1）通过 PHP 预定义类 mysqli 实现与 MySQL 数据库的连接，代码如下：

```
$conn=new mysqli("localhost","root","111","db_database09");
$conn->query("set names utf8");
```

（2）调用存储过程 pro_reg 实现将用户录入的注册信息保存到数据库，代码如下：

```
if($sql=$conn->query("call pro_reg('".$nc."','".$pwd."','".$email."','".$address."')")){
    echo "<script>alert('用户注册成功!');</script>";
}else{
    echo "<script>alert('用户注册失败!');</script>";
}
```

秘笈心法

心法领悟 465：使用 PDO 数据库抽象类执行存储过程。

实现本实例时，使用 MySQLi 扩展库执行存储过程，除使用该方法外，还可以使用 PDO 数据库抽象类执行存储过程，在实现本实例时，可以将保存用户注册信息的代码进行如下更改来演示使用 PDO 技术执行 MySQL 存储过程的方法。

```
$db = new PDO('mysql:host=localhost;dbname=db_database09', 'root', 'root');    //实例 PDO 类，连接 MySQL 数据库
$stmt = $db->prepare("call pro_reg(?, ?, ?, ?)");                               //建立查询的预处理句柄
$stmt->bindParam(1, $_POST['nc']);                                              //绑定参数
$stmt->bindParam(2, $_POST['pwd']);
$stmt->bindParam(3, $_POST['email']);
$stmt->bindParam(4, $_POST['address']);
$stmt->execute();                                                               //执行查询
```

说明： 使用 PDO 连接 MySQL 数据库时，连接字符串 mysql:host=localhost;dbname=db_database09 中内容的顺序可以颠倒。

9.4 MySQL 触发器

触发器在数据库系统开发过程中具有非常重要的作用，例如可以防止有害数据录入数据库、可以改变或取消 insert、update 和 delete 语句的执行及在一个会话中监听数据库中的数据的改变。

实例 466	创建 MySQL 触发器 光盘位置：光盘\MR\09\466	初级 趣味指数：★★★

实例说明

如果用户打算在数据库中通过触发器实现某一动作的监听，那么首先应该创建触发器。触发器是在命令提示符下创建的，如图 9.31 所示。

```
命令提示符 - mysql -uroot -proot
mysql> use db_database09;
Database changed
mysql> create table tb_test(t_id varchar(20),t_name varchar(20));
Query OK, 0 rows affected (0.08 sec)

mysql> select * from tb_test;
Empty set (0.01 sec)
mysql> delimiter //
mysql> create trigger test_tri
    -> before insert on tb_test
    -> for each row
    -> set new.t_name='mrsoft'//
Query OK, 0 rows affected (0.02 sec)
mysql> insert  into tb_test(t_id,t_name)values('mr0001','123')//
Query OK, 1 row affected (0.01 sec)

mysql> select * from tb_test//
+--------+--------+
| t_id   | t_name |
+--------+--------+
| mr0001 | mrsoft |
+--------+--------+
1 row in set (0.00 sec)
```

图 9.31　触发器创建过程

关键技术

MySQL 数据库中创建触发器的格式如下：

```
create trigger <触发器名称>
{ before | after}
{insert | update | delete}
on <表名>
for each row
<触发器 SQL 语句>
```

参数说明：

create trigger <触发器名称>：创建一个新触发器，并指定触发器的名称。

{ before | after}：用于指定在 insert、update 或 delete 语句执行前触发还是在语句执行后触发。

on <表名>：用于指定响应该触发器的表名。

for each row：触发器的执行间隔。for each row 通知触发器每隔一行执行一次动作，而不是对整个表执行一次。

<触发器 SQL 语句>：触发器要执行的 SQL 语句。

设计过程

（1）在命令提示符下创建数据表 tb_test，代码如下：

```
create table tb_test(t_id varchar(20),t_name varchar(20))
```

（2）将换行标记转换为“//”，代码如下：

```
delimiter //
```

（3）创建触发器，实现无论用户向表 tb_test 添加什么数据都使字段 t_name 的内容为“mrsoft”，代码如下：

```
create trigger test_tri
before insert on tb_test
for each row
set new.t_name='mrsoft'
```

（4）向表 tb_test 中添加一条记录，并查看结果，代码如下：

```
insert into tb_test(t_id,t_name) values('mr0001', '123')//
select * from tb_test
```

秘笈心法

心法领悟 466：触发器中执行多条 SQL 语句。

如果该触发器要执行多条 SQL 语句，要将多条语句放在 begin...end 块中。

实例 467	查看 MySQL 触发器 光盘位置：光盘\MR\09\467	初级 趣味指数：★★★

实例说明

在 MySQL 数据库中创建触发器前应先查看数据库中的触发器，这样既可以使开发人员对指定的数据库中的所有触发器及其功能有一个直观的把握，而且可以避免创建同名或类似功能的触发器。运行本实例，如图 9.32 所示，首先在图中的文本框中输入要查看触发器的数据库名称，然后单击“查看”按钮，即可将该数据库中所有触发器的详细信息显示出来。



图 9.32　查看触发器

关键技术

查看 MySQL 数据库中触发器的详细信息，可以通过如下语句实现：



删除相关 id 的工作后，还没来得及删除学生信息表中该学生的信息时就发生停电等意外，则再重新查找该学生的成绩时是无法查找到的。

本实例将采用事务处理方式，对学生信息表和学生成绩表中的数据进行删除。运行本实例，学生信息及学生成绩信息分别如图 9.33 和图 9.34 所示。当删除图 9.33 中的学生信息后，查看学生成绩信息时可以发现，与该学生对应的成绩也被全部删除。

图 9.33　查看学生信息

图 9.34　查看学生成绩

关键技术

事务的处理可以通过 PHP 的预定义类 mysqli 的以下方法实现。

autocommit(boolean)：该方法用于限定查询结果是否自动提交，如果参数为 true 则自动提交，如果参数为 false 则关闭自动提交。MySQL 数据库默认为自动提交。

rollback()：利用 mysqli 类中的该方法可以实现事务的回滚。

commit()：利用该方法可以实现提交所有查询。

设计过程

（1）建立数据库及数据表并实现与数据的连接，代码如下：

```
$conn=new mysqli("localhost","root","111","db_database09");
$conn->query("set names utf8");
```

（2）显示所有学生的基本信息，代码如下：

```
<?php
include_once("conn.php");
$sql=$conn->query("select * from tb_stu");
$info=$sql->fetch_array(MYSQLI_ASSOC);
if($info==NULL){
      echo "暂无学生信息!";
}else{
      do{
?>
      <tr>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sname];?></div></td>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sno];?></div></td>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[sage];?></div></td>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[saddress];?></div></td>
            <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo $info[ssfzh];?></div></td>
            <td bgcolor="#FFFFFF"><div align="center"><a href="javascript:if(window.confirm('确定删除该学生信息
么?')==true){window.location.href='delete.php?id=<?php echo $info[id];?>';}">删除</a></div></td>
      </tr>
<?php
      }while($info=$sql->fetch_array(MYSQLI_ASSOC));
}
?>
```

在实现该模块功能时，可以利用 JavaScript 实现该页与 delete.php 页面的信息传递，代码如下：

```
<a href="javascript:if(window.confirm('确定删除该学生信息么?')==true){window.location.href='delete.php?id=<?php echo $info[id];?>';}">删除</a>
```

window 对象的 confirm()方法用于弹出一个对话框，提示用户是否真正删除某学生的所有信息。

（3）利用事务处理机制对学生的基本信息进行删除，代码如下：

```
$id=$_GET[id];
include_once("conn.php");
$conn->autocommit(false);
if(!$conn->query("delete from tb_sco where id='".$id."'")){
```

```
        $conn->rollback();
    }
    if(!$conn->query("delete from tb_stu where id='".$id."'")){
        $conn->rollback();
    }
    $conn->commit();
    $conn->autocommit(true);
    echo "<script>window.location.href='index.php';</script>";
```

秘笈心法

心法领悟 468：使用 PDO 数据库抽象类执行批处理事务。

本实例使用 MySQLi 扩展库实现对 MySQL 数据库事务的处理，而使用 PDO 数据库抽象类同样可以实现此功能，其示例代码如下所示：

```
try {
    $db = new PDO('mysql:host=localhost;dbname=db_database09', 'root', 'root');    //实例 PDO 类，连接 MySQL 数据库
    $db->beginTransaction();                                                       //开启事务
    $db->query($sql1);                                                             //执行查询 1
    $db->query($sql2);                                                             //执行查询 2
    $db->commit();                                                                 //提交查询
} catch (PDOException $e) {
    $db->rollBack();                                                               //事务回滚
    echo $e->getMessage();
}
```

上述代码中，首先使用 new 关键字对 PDO 类进行实例化并返回 PDO 对象，并使用 PDO 对象的 beginTransaction()方法开启事务，然后执行多条查询语句，最后使用 PDO 对象的 commit()方法提交事务。在上述操作中使用 try...catch 语句判断是否发生了异常，如果是则调用 PDO 对象的 rollBack()方法对事务进行回滚。



实例 469　使用事务处理技术实现银行的安全转账

光盘位置：光盘\MR\09\469

初级

趣味指数：★★★

实例说明

在实现银行转账过程中，发生意外是在所难免的，为了避免因意外而造成不必要的损失，在银行转账时经常使用事务处理方式。运行本实例，如图 9.35 所示，在图中的文本框中输入要转给 B 账户的金额后，单击“转账”按钮即可实现转账。

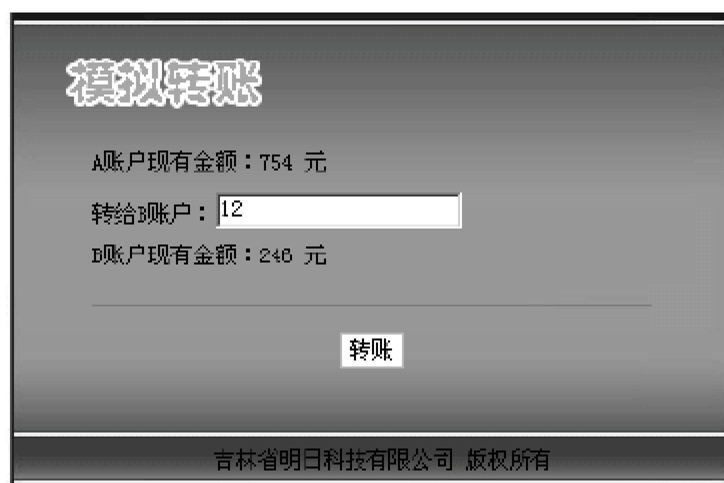


图 9.35　银行转账

关键技术

本实例的关键技术是如何应用 PHP 的事务处理机制处理转账过程中可能遇到的意外。应用事务实现转账的代码如下：

```
$tob=$_POST[tob];                                      //接收提交的转账金额
include_once("conn.php");                              //包含数据库连接文件 conn.php
```

```
$conn->autocommit(false);                                                        //取消查询自动提交
if(!$conn->query("update tb_zy set money=money-'".$tob."' where flag='mrsoft'")){    //减少指定账户所要转账的金额
    $conn->rollback();                                                           //如果失败则事务回滚
}
if(!$conn->query("update tb_zy set money=money+'".$tob."' where flag='mr'")){        //增加指定账户所要的转账金额
    $conn->rollback();                                                           //如果失败则事务回滚
}
$conn->commit();                                                                 //提交查询
$conn->autocommit(true);                                                         //恢复查询自动提交
echo "<script>window.location.href='index.php';</script>";                       //重定向到首页
```

首先调用 mysqli 类的 autocommit()方法关闭数据库的自动提交，然后减少 A 账户一定数量的金额并判断查询是否顺利执行，如果在此过程中发生意外，则通过调用 mysqli 类的 rollback()方法回滚，不做任何处理，反之为 B 账户增加相同的金额。最后通过调用 mysqli 类的 commit()方法提交查询，实现转账。

设计过程

（1）利用 MySQLi 扩展技术实现与 MySQL 数据库的连接，代码如下：

```
$conn=new mysqli("localhost","root","111","db_database09");     //建立与 MySQL 数据库的连接
$conn->query("set names utf8");                                 //设置数据库字符集编码为 utf-8
```

（2）查询 A 账户与 B 账户的现有金额并在页面中显示，代码如下：

```
include_once("conn.php");                                        //包含数据库连接文件 conn.php
$sql=$conn->query("select * from tb_zy where flag='mrsoft'");    //查询 mrsoft 用户的信息
$info=$sql->fetch_array(MYSQLI_ASSOC);
$sql1=$conn->query("select * from tb_zy where flag='mr'");       //查询 mr 用户的信息
$info1=$sql1->fetch_array(MYSQLI_ASSOC);
```

（3）实现转账。该部分的关键代码已经在关键技术中给出，这里不再赘述。

秘笈心法

心法领悟 469：使用 MySQLi 扩展库管理 MySQL 数据库事务时，需要关闭事务的自动提交。

默认情况下使用 MySQLi 扩展库管理 MySQL 数据库时，事务是自动提交的，所以在对事务管理时首先需要关闭事务的自动提交，其方法如下：

```
$mysqli->autocommit(false);
```

从上述代码可知，通过调用 mysqli 类的 autocommit()方法，并为其参数指定 FALSE 值即可关闭事务的自动提交。同理，如果为该方法的参数指定 TRUE 值则可以恢复事务的自动提交功能。

第10章

PHP操作SQL Server数据库

- 连接数据库
- 操作数据库
- SQL Server数据库的综合应用
- SQL Server数据库的安全与维护

10.1　连接数据库

SQL Server 是 Microsoft 公司开发的数据库产品。随着 SQL Server 2005 的诞生，越来越多的企业开始使用 SQL Server，下面将通过具体实例讲解 PHP 连接 SQL Server 数据库的方法。

实例 470	通过 ADO 方式连接 SQL Server 数据库 光盘位置：光盘\MR\10\470	高级 趣味指数：★★★★

实例说明

本实例通过 ADO 方式连接 SQL Server 数据库，运行本实例，如果出现如图 10.1 所示的用户信息页面，说明数据库连接成功。

用户信息	
用户名	密　码
mr	mrsoft

图 10.1　ADO 方式连接 SQL Server 数据库

关键技术

PHP 通过预定义类 com 来使用 ADO 方法操作数据库，该类的详细说明如下：

```
string com::com( string module_name [, string server_name [, int codepage]])
```

参数说明：

module_name：被请求组件的名字或 class-id。

server_name：DCOM 服务器的名字。

codepage：指定用于将 PHP 字符串转换成 UNICODE 字符串的代码页，反之亦然。该参数的取值有 CP_ACP、CP_MACCP、CP_OEMCP、CP_SYMBOL、CP_THREAD_ACP、CP_UTF7 和 CP_UTF8。

本实例与 SQL Server 建立连接的代码如下：

```
$conn = new com("adodb.connection");
$connstr = "provider = sqloledb;data source=PC-201006101638;uid=sa;pwd=;database=db_database10";
$conn -> open($connstr);
```

设计过程

（1）在 SQL Server 中创建数据库及数据表用于连接测试。

（2）连接 SQL Server 数据库，建立记录集，通过 while 语句完成用户信息的输出，代码如下：

```
<?php
if($_GET[id] == '1'){
        $sql = "select * from tb_demo01";
        $rs = new com("adodb.recordset");
        $rs -> open($sql,$conn,1,1);
        if($rs -> eof || $rs -> bof){
            echo "暂无用户注册信息";
        }else{
?>
                <tr>
                    <td align="center">用户名</td>
                    <td align="center">密  码</td>
                </tr>
```



参数说明：

string query：SQL 语句。

link_identifier：连接标识。

（3）mssql_fetch_row()函数，取得一行数据，保存于数组中，其语法如下：

```
array mssql_fetch_row ( resource result )
```

参数说明：

resource result：结果集。

本实例与 SQL Server 建立连接的代码如下：

```
$conn = mssql_connect("localhost","sa","") or die ("Connect MySQL False");
mssql_select_db("db_database10",$conn);
```

设计过程

在本实例中，通过 mssql_connect()函数连接 SQL Server，应用 mssql_query()函数执行查询语句，应用 mssql_fetch_row()函数和 while 语句完成用户信息的循环输出，其关键代码如下：

```
<?php
	$conn = mssql_connect ( "localhost", "sa", "" ) or die ( "Connect SQL Server False" );		//连接 SQL Server 服务器
	mssql_select_db ( "db_database10", $conn );		//连接 db_database10 数据库
	if ($_GET [id] == '1') {
?>
<?php
		if($rs = mssql_query("select * from tb_demo01")){		//如果返回结果集
			while($rst = mssql_fetch_row($rs)){		//循环输出结果集
?>
		<tr>
			<td bgcolor="#FFFFFF" align="center"><?php echo $rst[0];?></td>
			<td bgcolor="#FFFFFF" align="center"><?php echo $rst[1];?></td>
		</tr>
<?php
			}
		}else{
			echo "<script>alert('数据表存在问题');</script>";
		}
	}
?>
```

秘笈心法

心法领悟 471：如何检测 PHP 是否支持 SQL Server 数据库。

编写脚本文件，输入如下代码：

```
<?php
	echo phpinfo();
?>
```

经 IE 浏览器运行本文件，如果网页中显示 mssql 内容，表示 PHP 支持 SQL Server 数据库。

如果用户使用的 SQL Server 数据库是 2000 版本，可能会出现 unable connect server:localhost 错误，造成这种错误的原因一般分为两种情况：

（1）代码存在错误。检查 SQL Server 服务管理器是否处于正在运行的状态。

（2）系统可能缺少 ntwdblib.dll 文件。用户可以手动下载 ntwdblib.dll-2000.80.2039.0 文件，将此文件复制到 Apache\bin 文件夹和 php\bin 文件夹下，然后重启 SQL Server 服务器和 Apache 服务器即可。也可为 SQL Server 打上 sp4 补丁，然后重启服务器。

10.2　操作数据库

SQL Server 数据库与 MySQL 数据库虽然都是存储信息的载体，但是在操作方法与函数上略有不同。本节

将具体讲述对 SQL Server 数据库的相关操作。

实例 472　通过 ADO 方式实现指定时间段的信息检索

光盘位置：光盘\MR\10\472

高级

趣味指数：★★★★

实例说明

在进行网站开发过程中，经常用到对指定时间段的信息检索，例如，检索指定时间段内有哪些 PHP 图书出版等。运行本实例，当单击“信息检索”超链接时，将显示 2009 年~2010 年明日科技编写的所有 PHP 图书，其运行效果如图 10.3 所示。

通过ADO方式实现指定时间段的信息检索

信息检索

图书ID	图书名称	图书价格	出版社
1	《PHP范例宝典》	78	人民邮电出版社
2	《PHP实战宝典》	74	清华大学出版社
3	《PHP网络编程》	65	东财大学出版社

图 10.3　通过 ADO 实现指定时间段的信息检索

关键技术

本实例通过 ADO 操作 SQL Server 数据库，执行指定时间段内的信息查询操作，其信息检索的 SQL 语句代码如下：

```
$sql = "select id,bname,price,pub from tb_demo03 where date>='2009-01-01' and date<='2010-12-31'";
```

设计过程

创建 index.php 文件，首先通过 ADO 连接 SQL Server 数据库，然后建立记录集，实现用户信息的循环输出，其关键代码如下：

```
<?php
$conn = new com ( "adodb.connection" );
$connstr = "provider = sqloledb;data source=PC-201006101638;uid=sa;pwd=;database=db_database10";        //ADO 连接数据库
$conn->open ( $connstr );
if ($_GET [id] == "1") {
        $sql = "select id,bname,price,pub from tb_demo03 where date>='2009-01-01' and date<='2010-12-31'";
        $rs = new com ( "adodb.recordset" );
        $rs->open ( $sql, $conn );
        if ($rs->eof || $rs->bof) {
                echo "数据库暂无信息";
        } else {
                ?>
        <tr>
                <td align="center">图书 ID</td>
                <td align="center">图书名称</td>
                <td align="center">图书价格</td>
                <td align="center">出版社</td>
        </tr>
<?php
                while ( ! $rs->eof ) {
                        ?>
        <tr>
                <td align="center" bgcolor="#FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rs->fields ( id )->value ); ?></td>
                <td align="center" bgcolor="#FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rs->fields ( bname )->value ); ?></td>
                <td align="center" bgcolor="#FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rs->fields ( price )->value ); ?></td>
                <td align="center" bgcolor="#FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rs->fields ( pub )->value ); ?></td>
        </tr>
```

```
<?php
                $rs -> movenext;
                }
        }
}
?>
```

秘笈心法

心法领悟 472：解决页面乱码问题。

由于在 SQL Server 中使用的编码格式是 gbk，而页面中的编码格式为 uft-8，所以需使用转码函数 iconv()对输出结果进行转码，否则将输出乱码。

实例 473　检索商品信息　高级

光盘位置：光盘\MR\10\473

趣味指数：★★★★

实例说明

在实际开发中，很多程序都会涉及到对信息的检索。运行本实例，输入商品 ID，单击“检索”按钮，将查找出匹配的商品信息，并且对商品名称描红显示，运行结果如图 10.4 所示。

图 10.4　商品信息的检索

关键技术

本实例通过 mssql_connect()函数连接 SQL Server 服务器，通过 mssql_select_db()函数连接 db_database10 数据库，通过 mssql_query()函数执行查询语句，通过 mssql_fetch_row()函数获取查询结果中的数据，其关键代码如下：

```
<?php
$conn = mssql_connect ( "localhost", "sa", "" ) or die ( "Connect SQL Server False" );          //连接数据库服务器
mssql_select_db ( "db_database10", $conn ) or die ( "Connect Database False" );                //连接数据库
if ($_POST [sub]) {                                                                            //判断提交按钮是否为空
        if ($_POST [text] == "" || $_POST [text] == "输入商品 ID") {                            //如果提交 ID 值为空，则给出提示信息
                echo "<script>alert('请输入查询关键字');</script>";
        } else {
                $rs = mssql_query ( "select * from tb_demo04 where id=" . $_POST [text] );     //执行查询语句
?>
```

设计过程

（1）在 index.php 文件中，首先创建 form 表单，将商品的 ID 值提交到本页，其代码如下：

```
<form action="" method="post">
        <input class="one" type="text" name="text" value="输入商品 ID" size="15" onfocus="this.value=''" />
        <input class="two" type="submit" name="sub" value=" " />
</form>
```

（2）在 index.php 文件中，通过 mssql_connect()函数连接 SQL Server，根据 form 表单中提交的 ID 值执行查询语句，并循环输出查询结果，代码如下：

```
<?php
$conn = mssql_connect ( "localhost", "sa", "" ) or die ( "Connect SQL Server False" );
```

```
mssql_select_db ( "db_database10", $conn ) or die ( "Connect Database False" );
if ($_POST [sub]) {
	if ($_POST [text] == "" || $_POST [text] == "输入商品 ID") {
		echo "<script>alert('请输入查询关键字');</script>";
	} else {
		$rs = mssql_query ( "select * from tb_demo04 where id=" . $_POST [text] );
		?>
	<tr>
		<td align="center">商品 ID</td>
		<td align="center">商品名称</td>
		<td align="center">商品价格</td>
		<td align="center">商品类型</td>
	</tr>
<?php
	$rst = mssql_fetch_row ( $rs );

?>
	<tr>
		<td align="center" bgcolor="#FFFFFF"><?php echo $rst [0]; ?></td>
		<td ><font size="+1" color="#FF0000"><b><?php echo iconv ( 'gbk', 'utf-8', $rst [1] );?></b></font></td>
		<td bgcolor="#FFFFFF" align="center"><?php echo iconv ( 'gbk', 'utf-8', $rst [2] ); ?></td>
		<td bgcolor="#FFFFFF" align="center"><?php echo iconv('gbk','utf-8',$rst[3]);?></td>
	</tr>
<?php
	}
}
?>
```

秘笈心法

心法领悟 473：判断连接 SQL Server 数据库是否成功的方法。

判断连接数据库是否成功的方法有很多。例如，可以应用 if()条件语句对函数进行判断。但是笔者喜欢用 or die()方法实现对目标函数的检测，其最大的优点是书写简单、逻辑清晰、代码量小。

实例 474 获取指定位置的图书信息

光盘位置：光盘\MR\10\474

高级

趣味指数：★★★★

实例说明

在使用 SQL Server 数据库时，很多时候需要获取指定的第 n 条数据信息。用 MySQL 数据库实现此功能非常简单。那么如何在 SQL Server 数据库中实现相同的功能呢？下面就来向读者讲解实现此功能的具体方法。运行本实例，在图 10.5 所示的文本框中输入指定的 ID 值，单击“确定”按钮，将输出如图 10.6 所示的内容。

获取指定位置的图书信息

请输入数字 确定

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00

图 10.5 数据库信息

获取指定位置的图书信息

2 确定

图书ID	图书名称	图书价格	出版日期
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00

图 10.6 获取指定位置的图书信息

关键技术

本实例主要应用 mssql_data_seek()函数。mssql_data_seek()函数将指定的结果标识所关联的 SQL Server 结果



图 10.7　显示数据库中数据

图 10.8　获取字段信息

关键技术

mssql_fetch_field()函数，从结果集中获取列信息并作为对象返回，其语法如下：

```
object mssql_fetch_field ( resource result [, int field_offset] )
```

对象的属性如表 10.1 所示。

表 10.1　mssql_fetch_field()函数返回的对象的属性

属　性	说　明
name	列名
column_source	具体的列名
max_length	该列最大长度
numeric	1，如果该列是 numeric
type	该列的类型

设计过程

（1）新建 conn.php 文件，通过 mssql 函数库连接 SQL Server 数据库，其代码如下：

```
<?php
$conn = mssql_connect("localhost","sa","") or die ("连接 SQL Server 时发生错误");
mssql_select_db("db_database10",$conn) or die ("连接数据库发生错误");
?>
```

（2）新建 index.php 文件，首先，创建 form 表单，提交定位查询所使用的数字。然后，对表单中提交的定位查询数据进行判断，判断其值是否为空、提交的数据是否是数字、提交的值是否超出范围，如果完全符合条件，则应用 mssql_fetch_field()函数获取表单中指定字段的数据；如果判断结果不符合要求，则给出相应的提示信息，并且循环输出数据表中所有的数据，其关键代码如下：

```
<?php
include_once ("conn.php");
$rs = mssql_query ( "select * from tb_demo10" );
if (isset ( $_POST [sub] )) {
 if ($_POST [text] == "" || $_POST [text] == "请输入数字") {
        echo "<script>alert('请在文本框中输入信息');</script>";
 } else {
        if (! preg_match ( "/\d/", $_POST [text] )) {
                echo "<script>alert('请输入数字')</script>";
        } else {
                if ($_POST [text] > 3) {
                        echo "<script>alert('超出取值范围');</script>";
                } else {
                        $ts = @mssql_fetch_field ( $rs, $_POST [text] );
                }
                ?>
<table bgcolor="#DEED88" width="580px">
        <tr>
                <td class="o">name</td>
```

```
                <td class="o">column_source</td>
                <td class="o">type</td>
                <td class="o">max_length</td>
        </tr>
        <tr>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts->name )?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts->column_source )?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts->type )?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $ts->max_length )?></td>
        </tr>
<?php
        }
}
} else {
?>
```

秘笈心法

心法领悟 475：使用 mssql_fetch_field()函数的注意事项。

mssql_fetch_field()函数可以用来从某个查询结果中获取字段的信息。如果没有指定字段偏移量，则下一个尚未被 mssql_fetch_field()函数获取的字段被提取。

实例 476　利用 mssql_field_name()函数获取字段名称

光盘位置：光盘\MR\10\476

高级

趣味指数：★★★★

实例说明

实例 475 中通过 mssql_fetch_field()函数返回对象的 name 属性获取字段的名称，下面将介绍在 mssql 函数库中专门用来获取字段名称的方法。运行本实例，在文本框中输入一个有效的字段值，单击“确定”按钮，将输出如图 10.9 所示的页面。

利用mssql_field_name()获取字段名称

2　确定

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00
数据表字段名称为	price		

图 10.9　利用 mssql_field_name()函数获取字段名称

关键技术

本实例主要应用 mssql_field_name()函数获取字段的名称，其语法如下：

```
string mssql_field_name ( resource result, int field_index )
```

该函数返回指定字段索引的字段名。参数 result 必须是一个合法的结果标识符，参数 field_index 是该字段的数字偏移量。

设计过程

（1）新建 conn.php 文件，通过 mssql 函数库连接 SQL Server 数据库。

（2）新建 index.php 文件，首先，创建 form 表单，提交定位查询所使用的数字。然后，执行查询语句，并通过 while 语句和 mssql_fetch_row()函数循环输出查询结果集中的数据。最后，获取表单提交的数据，应用

mssql_field_name()函数获取指定字段的名称，其代码如下：

```
<?php
}
if (isset ( $_POST [sub] )) {
        if ($_POST [te] == "" || $_POST [te] == "输入数字") {
                echo "<script>alert('请在文本框中输入内容');</script>";
        } else {
                if (preg_match ( "/\d/", $_POST [te] )) {
                        if ($_POST [te] > 4) {
                                echo "<script>alert('输入的数字超出范围');</script>";
                        } else {
                                $rss = mssql_field_name ( $rs, $_POST [te] );
                        }
                } else {
                        echo "<script>alert('您输入的不是数字');</script>";
                }
        }
        ?>
<tr>
        <td bgcolor="#FFFFFF">数据表字段名称为</td>
        <td bgcolor="#FFFFFF" style="color: #FF0000"><?php echo $rss;?></td>
</tr>
<?php
        }
?>
```

秘笈心法

心法领悟 476：使用 mssql_field_name()函数的注意事项。

在使用此函数时，参数 field_index 从 0 开始。例如，第 3 个字段的索引值其实是 2，第 4 个字段的索引值是 3，依次类推。

实例 477　通过结果集对象输出数据

光盘位置：光盘\MR\10\477

高级

趣味指数：★★★★

实例说明

从结果集中获取数据的方法有很多，本实例主要讲解利用数据对象输出结果集中的数据。运行本实例如图 10.10 所示。

通过结果集对象输出数据

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00

图 10.10　通过结果集对象输出数据

关键技术

mssql_fetch_object()函数，从结果集中获取一行作为对象，如果没有更多行则返回 FALSE，其语法如下：

```
object mssql_fetch_object ( resource result )
```

该函数和 mssql_fetch_array()函数类似，只是返回一个对象而不是数组，即只能通过字段名来访问数据，而不是偏移量（数字是合法的属性名）。

设计过程

（1）新建 conn.php 文件，连接 SQL Server 数据库。

（2）创建 index.php 文件，包含数据库连接文件，通过 mssql_query()函数执行查询操作，通过 while 语句和 mssql_fetch_object()函数循环输出查询结果集中的数据，其代码如下：

```
<?php
while ( $rst = mssql_fetch_object ( $rs ) ) {
?>
<tr>
	<td style="background: #FFFFFF"><?php echo iconv ( "gbk", "utf-8", $rst->id );  ?></td>
	<td style="background: #FFFFFF"><?php echo iconv ( "gbk", "utf-8", $rst->bname ); ?></td>
	<td style="background: #FFFFFF"><?php echo iconv ( "gbk", "utf-8", $rst->price );?></td>
	<td style="background: #FFFFFF"><?php echo iconv ( "gbk","utf-8",$rst->date);?></td>
</tr>
<?php
	}
?>
```

秘笈心法

心法领悟 477：使用 mssql_fetch_object()函数的注意事项。

使用 mssql_fetch_object()函数时需要注意，此函数返回的字段名是区分大小写的。

实例 478	通过结果集数组输出数据 光盘位置：光盘\MR\10\478	高级 趣味指数：★★★★

实例说明

实例 477 中讲解了通过结果集对象输出数据，下面将通过另外一个函数将结果集以数组的形式输出。运行本实例，结果如图 10.11 所示。

通过结果集数组输出数据

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00

图 10.11　通过结果集数组输出数据

关键技术

本实例主要应用 mssql_fetch_assoc()函数，从结果集中获取一行作为关联数组。如果没有更多行则返回 FALSE，其语法如下：

```
array mssql_fetch_assoc ( resource result_id )
```

参数 result_id 为指定的结果集。

设计过程

（1）新建 conn.php 文件，连接 SQL Server 数据库。

（2）创建 index.php 文件，包含数据库连接文件，通过 mssql_query()函数执行查询操作，通过 while 语句和 mssql_fetch_assoc()函数循环输出查询结果集中的数据，其代码如下：

```
<?php
        include_once ("conn.php");
        $rs = mssql_query ( "select * from tb_demo10" );
?>
<table width="580px" bgcolor="#FC8C7E">
        <tr>
                <td style="color: #FFFFFF; text-align: center">图书 ID</td>
                <td style="color: #FFFFFF; text-align: center">图书名称</td>
                <td style="color: #FFFFFF; text-align: center">图书价格</td>
                <td style="color: #FFFFFF; text-align: center">出版日期</td>
        </tr>
<?php
        while ( $rst = mssql_fetch_assoc ( $rs ) ) {
?>
        <tr>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [id] ); ?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [bname] ); ?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [price] );  ?></td>
                <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [date] ); ?></td>
        </tr>
<?php
        }
?>
```

秘笈心法

心法领悟 478：使用 mssql_fetch_assoc()函数的注意事项。

由于本函数返回的是一个关联数组，所以在获取数组中的元素时必须使用字段名称作为数组的索引，同时还要注意，其数组的索引是区分大小写的。



实例 479　返回结果集中的单元内容　高级

光盘位置：光盘\MR\10\479　趣味指数：★★★★

实例说明

如果用户只想查看用户名或者只想查看单一的一项数据，程序员就没有必要在与数据库的交互中取得一行数据，只需要从数据库中取得单一字段信息即可。本实例通过函数 mssql_result()实现获取结果数据，运行结果如图 10.12 所示。



图 10.12　返回结果集中的单元内容

关键技术

本实例的关键点是如何运用 mssql_result()函数返回结果集中的单元内容，其语法如下：

```
string mssql_result ( resource result, int row, mixed field )
```

参数说明：

resource result：结果集。

int row：列数。

mixed field：指定字段。

设计过程

（1）新建 conn.php 文件，连接 SQL Server 数据库。

（2）创建 index.php 文件，首先，创建 form 表单，提交指定要查询的列数和字段名称。然后，包含数据库连接文件，通过 mssql_query()函数执行查询操作，通过 while 语句和 mssql_fetch_row()函数循环输出查询结果集中的数据。接着，对 form 表单中提交的数据进行判断，如果符合要求则运用 mssql_result()函数获取结果集中指定单元的内容，其关键代码如下：

```
<?php
	}
	if (isset ( $_POST [sub] )) {
		$text = $_POST [te];
		$select = $_POST [select];
		if ($_POST [te] == "" || $_POST [te] == "输入数字") {		//判断输入的值是否为空
			echo "<script>alert('文本框不能为空')</script>";
		} else {
			if (preg_match ( "/\d/", $text )) {				//判断输入的值是否是数字
				if ($text < 5) {
					$rst = mssql_result ( $rs, $text, $select );	//获取指定单元的内容
					echo iconv ( 'gbk', 'utf-8', $rst );		//输出指定单元的内容
				} else {
					echo "<script>alert('文本框超出取值范围');</script>";
				}
			} else {
				echo "<script>alert('文本框输入的不是数字');</script>";
			}
		}
	}
```

秘笈心法

心法领悟 479：使用 mssql_result()函数的注意事项。

在使用此函数的 int rows 参数时要注意下标是以 0 开始的，如在页面中输入数字 2，其实是表示第 1 个数据单元。

实例 480　利用函数返回指定字段的长度

光盘位置：光盘\MR\10\480

高级

趣味指数：★★★★

实例说明

在创建数据表时，需要定义字段的最大长度，如果插入的数据信息超过这个最大长度，字段就不能完整地输出。运行本实例，通过函数自动获取字段信息的长度，结果如图 10.13 所示。

图 10.13　返回指定字段的长度

关键技术

获取指定字段的长度应用的是 mssql_field_length()函数，其语法如下：

```
int mssql_field_length ( resource result, int field_offset )
```

参数说明：

resource result：结果集。

int field_offset：字段数。

设计过程

（1）新建 conn.php 文件，连接 SQL Server 数据库。

（2）在 index.php 文件中，创建 form 表单，提交指定要查询的字段值，并且将数据提交到本页。

（3）在 index.php 文件中，包含数据库连接文件，执行查询语句，完成数据表中数据的循环输出，并对表单中提交的值进行判断，如果符合要求，则应用 mssql_field_length()函数获取表单提交的字段的长度值，并且输出，其关键代码如下：

```
<?php
	include_once ("conn.php");
	$rs = mssql_query ( "select * from tb_demo10" );
	?>
		<table width="580px" bgcolor="#FEF1BA">
			<tr>
				<td style="color: #FFFFFF; text-align: center">图书 ID</td>
				<td style="color: #FFFFFF; text-align: center">图书名称</td>
				<td style="color: #FFFFFF; text-align: center">图书价格</td>
				<td style="color: #FFFFFF; text-align: center">出版日期</td>
			</tr>
	<?php
	if (isset ( $_POST [sub] )) {
		$rss = mssql_field_length ( $rs, $_POST [select] );
		echo "<font size='+1' color='red'>字段长度为" . $rss . "</font>";
	}
	while ( $rst = mssql_fetch_row ( $rs ) ) {
		?>
		<tr>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [0] ); ?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [1] ); ?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [2] ); ?></td>
			<td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [3] ); ?></td>
		</tr>
<?php
	}
?>
```

秘笈心法

心法领悟 480：获取数据表中指定字段的类型。

在 mssql 函数库中，不但可以获取到数据表中指定字段的名称、长度、定位指针位置，而且可以获取到字段的类型，其应用的是 mssql_field_type()函数。

实例 481	利用函数返回指定字段的类型 光盘位置：光盘\MR\10\481	高级 趣味指数：★★★★

实例说明

通过上面的实例，相信读者已经掌握了如何利用函数获取字段的长度。下面将讲解如何利用函数获取指定

字段的字段类型。运行本实例，如图 10.14 所示。

图 10.14　返回指定字段的类型

关键技术

本实例主要应用 mssql_field_type()函数，获取结果集中指定字段的类型，其语法如下：

```
string mssql_field_type ( resource result [, int offset] )
```

参数说明：

resource result：结果集。

int field_offset：字段数。

设计过程

（1）新建 conn.php 文件，连接 SQL Server 数据库。

（2）在 index.php 文件中，创建 form 表单，提交指定要查询的字段值，并且将数据提交到本页，其代码如下：

```
<form action="" method="post">
        <select name="select">
          <option value="0">ID 字段</option>
          <option value="1">名称字段</option>
          <option value="2">价格字段</option>
          <option value="3">日期字段</option>
      </select>
        <input type="submit" name="sub" value="" class="two" />
</form>
```

（3）在 index.php 文件中，包含数据库连接文件，执行查询语句，完成数据表中数据的循环输出，并对表单中提交的值进行判断，如果符合要求，则应用 mssql_field_type ()函数获取表单提交的字段的类型值，并且输出，其关键代码如下：

```
<?php
        include_once ("conn.php");
        $rs = mssql_query ( "select * from tb_demo10" );
        ?>
        <?php
        if (isset ( $_POST [sub] )) {
                $rss = mssql_field_type ( $rs, $_POST [select] );
                echo "<font size='+1' color='red'>字段类型为" . $rss . "</font>";
        }
        while ( $rst = mssql_fetch_row ( $rs ) ) {
                ?>
                <tr>
                        <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [0] ); ?></td>
                        <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [1] ); ?></td>
                        <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [2] ); ?></td>
                        <td style="background: #FFFFFF"><?php echo iconv ( 'gbk', 'utf-8', $rst [3] ); ?></td>
                </tr>
<?php
        }
?>
```

秘笈心法

心法领悟 481：SQL Server 数据库中的字段类型。

SQL Server 数据库中的数据类型与 MySQL 数据库中的数据类型有所不同，SQL Server 中的字段类型包括 int、real、string、blob、date 等。

实例 482　取得结果集中的字段数目　高级

光盘位置：光盘\MR\10\482

趣味指数：★★★★

实例说明

在 SQL Server 数据库中，取得结果集中的字段数目其实与操作 MySQL 数据库大同小异。下面就来讲述如何在 SQL Server 数据库中取得结果集中的字段数目，运行结果如图 10.15 所示。

取得结果集中的字段数目

获取字段数目

当前数据表字段数目为4

图书ID	图书名称	图书价格	出版日期
1	《PHP开发实战宝典》	78元	2010/八月/31 0:00
2	《PHP标准教程》	79元	2010/八月/31 0:00
3	《PHP网络编程宝典》	73元	2010/八月/31 0:00
4	《PHP范例宝典》	98元	2010/八月/31 0:00
5	《PHP范例手册》	68元	2010/八月/31 0:00

图 10.15　获取结果集的字段数目

关键技术

获取结果集的字段数目主要应用 mssql_num_fields()函数，其语法如下：

```
int mssql_num_fields( resource result )
```

参数 resource result 指结果集。

设计过程

（1）新建 conn.php 文件，通过 mssql_connect()函数连接 SQL Server 数据库服务器，然后通过 mssql_select_db()函数连接 db_database10 数据库，其代码如下：

```
<?php
$conn = mssql_connect("localhost","sa","") or die ("连接 SQL Server 时发生错误");
mssql_select_db("db_database10",$conn) or die ("连接数据库发生错误");
?>
```

（2）创建 index.php 脚本文件，包含数据库连接文件，通过 mssql_query()函数执行查询语句，应用 mssql_num_fields()函数获取查询结果集中总的字段数，并且通过 while 语句和 mssql_fetch_row()函数完成查询结果集中数据的循环输出，代码如下：

```
<?php
	include_once ("conn.php");
	$rs = mssql_query ( "select * from tb_demo10" );
	if ($_GET [id] == 1) {
		$rss = mssql_num_fields ( $rs );
		echo "<font size='+1' color='red'>当前数据表字段数目为" . $rss . "</font>";
	}
	?>
		<table width="580px" bgcolor="#CCE587">
				<tr>
```



```
        <?php
        while($rst = mssql_fetch_row($rs)){                                                    //循环输出
?>
<tr>
        <td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[0]);?></td>
        <td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[1]);?></td>
        <td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[2]);?></td>
        <td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[3]);?></td>
</tr>
<?php
        }
        if($_GET[id] == 1){                                                                    //当按钮被单击
                $rss = mssql_num_rows($rs);                                                    //返回结果
?>
<tr><td bgcolor="#FFFFFF">数据表数据</td><td bgcolor="#FFFFFF">信息数量为</td><td bgcolor="#FFFFFF"><?php echo $rss;?></td><td
bgcolor="#FFFFFF"></td></tr>
<?php
        }
?>
```

由于 SQL Server 数据库中使用的是 gbk 编码格式，而本实例的页面应用的是 utf-8 编码，所以在输出数据库中的数据时需要应用 iconv()转码函数对数据库的编码格式进行转换，否则将输出乱码。

秘笈心法

心法领悟 483：使用 mssql_num_rows()函数的注意事项。

mssql_num_rows()函数仅对 select 语句有效。

实例 484 mssql_free_result()函数的应用

光盘位置：光盘\MR\10\484

高级

趣味指数：★★★★

实例说明

在输出数据信息后要及时释放结果内存以节省内存空间。也许用户会觉得，不释放结果内存并没有什么严重影响。这是因为用户本机上运行的实例，数据结果很小，对于计算机运算速度来说没有太大的影响。但是如果结果集中包含大量的数据，计算机的运算速度就会明显变慢。运行本实例，通过 mssql_free_result()函数释放结果内存，如图 10.17 所示。



图 10.17　释放结果内存

关键技术

本实例主要应用 mssql_free_result()函数释放结果内存，其语法如下：

```
bool mssql_free_result ( resource result )
```

该函数将释放所有与结果标识符 result 相关联的内存。

设计过程

创建 index.php 脚本文件，包含数据库连接类，对类进行实例化操作，完成与 db_database10 数据库的连接，并循环输出数据表中的数据，当单击“释放结果内存”超链接时，执行释放结果集的操作，代码如下：

```
<?php
        include_once("conn.php");                                              //包含数据库文件
        $ms = new Mssql("localhost","sa","","db_database10");                  //实例化对象
        $rs = mssql_query("select * from tb_demo10");
?>
<?php
        while($rst = mssql_fetch_row($rs)){                                    //循环输出
?>
<tr>
        <td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[0]);?></td>
        td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[1]);?></td>
        <td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[2]);?></td>
        <td bgcolor="#FFFFFF"><?php echo iconv('gbk','utf-8',$rst[3]);?></td>
</tr>
<?php
        }
        if($_GET[id] == 1){
                $ms -> free_result($rs);                                       //释放结果集
        }
?>
```

秘笈心法

心法领悟 484：详解 mssql_free_result()函数。

mssql_free_result()函数将释放所有与结果标识符 result 相关联的内存，仅需要在考虑到返回很大的结果集时会占用较多内存时调用，在脚本结束后所有关联的内存都会被自动释放。

实例 485	mssql_close()函数的应用 光盘位置：光盘\MR\10\485	高级 趣味指数：★★★★

实例说明

在程序运行结束后，不但要释放结果内存，还要关闭与数据库的连接以节省系统资源。运行本实例，单击“关闭连接”超链接，将输出提示，运行结果如图 10.18 所示。



图 10.18　关闭与数据库的连接

关键技术

本实例主要是通过编写数据库类并通过对象句柄调用 mssql_close()函数实现关闭数据库连接的操作。其中



Server 数据库的连接，其中通过两种方法连接数据库：通过 mssql_connect()函数实现与数据库的暂时连接、通过 mssql_pconnect()函数实现与数据库的永久连接。Mssql 类的代码如下：

```
<?php
class Mssql{
	var $host;
	var $user;
	var $pwd;
	var $conn;
	public function __construct($host,$user,$pwd,$conn){				//构造函数
		$this -> host = $host;
		$this -> user = $user;
		$this -> pwd = $pwd;
		$this -> conn = $conn;
		$this -> connect();
	}
	public function connect(){
		if($this -> conn == "pconn"){
			$this -> conn = mssql_pconnect($this -> host,$this -> user,$this -> pwd);
			echo "<script>alert('与 SQL Server 数据库建立永久连接');</script>";
		}else{
			$this -> conn = mssql_connect($this -> host,$this -> user,$this -> pwd);
			echo "<script>alert('与 SQL Server 数据库建立暂时连接');</script>";
		}
	}
}
?>
```

设计过程

（1）新建 conn.php 文件，编写数据库连接类，其完整代码请参考关键技术中的内容。

（2）新建 index.php 文件，创建 form 表单，提交连接数据库所使用的方式。然后，包含数据库连接类，执行类的实例化操作，并调用 connect()方法完成与数据库的连接，其关键代码如下：

```
<?php
	include_once("conn.php");						//包含数据库连接类
	if(isset($_POST[sub])){						//判断按钮是否被单击
		if($_POST[select] == ""){					//判断下拉列表框内容
			echo "<script>alert('请选择连接方式');</script>";
		}else{
			$ms = new Mssql("localhost","sa","",$_POST[select]);		//实例化对象
		}
	}
?>
```

秘笈心法

心法领悟 486：对两种连接方式的选择。

永久连接和暂时连接从功能上说都可以实现与数据库的连接操作。至于如何选择，一般都是程序员的个人爱好问题。笔者喜欢用暂时连接的方式，虽然在操作完数据库后须及时关闭与数据库的连接，但是在结构上较为鲜明。

实例 487	添加图书类的实现 光盘位置：光盘\MR\10\487	高级 趣味指数：★★★★

实例说明

在实际 Web 开发过程中，无论是前台还是后台经常会用到向数据库中添加信息这一模块，如果所用之处都重复编写代码，这会使整个项目的代码重用率大大降低。为了克服上述缺陷，可以通过一个数据信息添加类来

实现某项目中的所有信息的添加工作。运行本实例，图书添加页面及查看图书信息页面分别如图 10.20 和图 10.21 所示，首先在图 10.20 所示的表单中输入某图书的信息，单击“添加”按钮即可将图书信息添加到数据库中，同时可以在图 10.21 所示的页面中查看所有已经添加的图书信息。

图 10.20　添加图书信息

书名	作者	出版社	出版时间
《**经验技巧》	明日科技	人民邮电出版社	2007-01-03
《*数据库开发完全手册》	明日科技	人民邮电出版社	2006-12-03
《**范例宝典》	明日科技	人民邮电出版社	2006-01-03
dfd	dfd	fdfd	2003-12-02

图 10.21　浏览图书信息

关键技术

本实例的关键是如何编写通用的添加信息类，其实现的基本原理是：通过构造函数对要执行的 SQL 语句进行初始化，这样即使表中的字段数目不同也可以实现将图书信息添加到数据库中，addbook 类的代码如下：

```
<?php
class addbook {
 private $add_sql;                         //定义私有变量
 public function __construct($x) {         //定义构造函数
        $this->add_sql = $x;               //为变量赋值
 }
 public function add() {                   //定义添加数据的方法
        include_once ("conn.php");
        if (@mssql_query ( $this->add_sql, $conn )) {
               echo "<script>alert('新书添加成功!');history.back();</script>";
        } else {
               echo "<script>alert('新书添加失败!');history.back();</script>";
        }

 }
}
$bookname = $_POST [bookname];
$auto = $_POST [auto];
$pub = $_POST [pub];
$pubtime = $_POST [pubtime];
$sql = "insert into tb_demo08(bookname,auto,pub,pubtime)values('$bookname','$auto','$pub','$pubtime')";
$adbook = new addbook ( $sql );
$adbook->add ();
?>
```

该类定义一个私有的数据成员$add_sql 用于保存传入该类的查询语句，并通过构造函数__construct()实现对该私有成员的初始化，最后定义该类的成员函数 add()执行添加操作。

设计过程

（1）新建 conn.php 文件，完成与数据库的连接。

（2）新建 addbook.php 文件，创建 form 表单，提交要添加的图书信息，将图书信息提交到 savebook.php 文件中。添加图书信息的页面效果如图 10.20 所示。

（3）创建 savebook.php 文件，编写数据添加类，定义 addbook()方法执行数据的添加操作。在该文件中完成类的实例化操作，并通过返回的对象调用类中的 addbook()方法，根据表单中提交的数据定义 insert 添加语句，最终执行数据的添加操作，其代码请参考关键技术中的内容。

（4）新建 index.php 文件，循环输出数据表中存储的图书信息，其关键代码如下：



```
        public function Delete(){                                          //定义删除方法
                if(!$rs = @mssql_query($this -> sql)){                     //如果执行删除操作发生错误
                        echo "<script>alert('SQL 语句错误');</script>";     //JavaScript 提示
                }else{
                        echo "<script>alert('删除成功');location.href='index.php?id=1';</script>";
                }
        }
}
?>
```

设计过程

（1）新建 conn.php 文件，编写数据库连接类，完成与 SQL Server 数据库的连接，其代码如下：

```
<?php
class ConnDemo{                                                            //定义类名称
        var $host;                                                         //主机名称
        var $user;                                                         //用户名称
        var $pwd;                                                          //用户密码
        var $db;                                                           //数据库名称
        var $conn;                                                         //连接标识符
        var $tb;                                                           //数据表名称
        public function __construct($host,$user,$pwd,$db,$tb){             //构造函数
                $this -> host = $host;
                $this -> user = $user;
                $this -> pwd = $pwd;
                $this -> db = $db;
                $this -> tb = $tb;
                $this -> connect();
        }
        public function connect(){
                 $this -> conn = $conn;
                $this -> conn = mssql_connect($this -> host,$this -> user,$this -> pwd) or die ("Connect mssql false");
                mssql_select_db($this -> db,$this -> conn) or die ("Connect Database false");
        }
}
?>
```

（2）新建 delete.php 文件，编写数据库删除类，并继承数据库连接类，其代码如关键技术中所示。

（3）新建 index.php 文件。首先，创建 form 表单，提交图书的 ID 值，作为执行删除操作的条件。然后，包含数据删除类文件，执行 DeleteDemo 类的实例化操作，并且向类中传递参数。接着，执行查询语句，循环输出数据表中的数据。最后，根据 form 表单提交的值，通过类实例化返回的对象调用 Delete()方法执行删除操作，其关键代码如下：

```
<?php
include ("delete.php");
$dd = new DeleteDemo ( "localhost", "sa", "", "db_database10", "tb_demo09" );
if ($_GET [id] == "1") {
 $rst = mssql_query ( "select * from tb_demo09" );
?>
<form action="" method="post">
<input type="text" size="10" name="text" value="输入 ID 删除" onfocus="this.value=''" class="one" />
        <input type="submit" name="sub" value=" " class="two" />
</form>
<table width="580px" bgcolor="#FA7672">
        <tr>
                <td align="center">图书名称</td>
                <td align="center">作者</td>
                <td align="center">出版社</td>
                <td align="center">出版时间</td>
        </tr>
<?php
 while ( $rstt = mssql_fetch_row ( $rst ) ) {
        ?>
        <tr>
                <td bgcolor="#FFFFFF" align="center"><?php  echo iconv ( 'gbk', 'utf-8', $rstt [0] );?></td>
                <td align="center" bgcolor="#FFFFFF"><?php  echo iconv ( 'gbk', 'utf-8', $rstt [1] );?></td>
                <td align="center" bgcolor="#FFFFFF"><?php  echo iconv ( 'gbk', 'utf-8', $rstt [2] );?></td>
```

```
                <td align="center" bgcolor="#FFFFFF"><?php  echo iconv ( 'gbk', 'utf-8', $rstt [3] );?></td>
        </tr>
<?php
}
}
if ($_POST [sub]) {                                                   //判断提交按钮的值
        $te = $_POST[text];                                           //获取提交的 ID
        $dd -> SQL($te);                                              //调用 SQL 方法定义 delete 删除语句
        $dd -> Delete();                                              //调用 Delete()方法执行删除操作
}
?>
</td>
        </tr>
</table>
```

秘笈心法

心法领悟 488：类的继承。

少数面向对象的语言（如 C++、Smalltalk 等）支持多继承，即一个子类有多个父类。但是 PHP 自身不支持多继承，只支持单继承，即一个子类只有一个父类。

实例 489　更新图书类的实现

光盘位置：光盘\MR\10\489

高级

趣味指数：★★★★

实例说明

实现更新图书信息类与删除图书信息类大同小异，只是将删除图书信息类中的两个核心方法替换为更新操作的方法即可。运行本实例，执行图书信息的更新操作，其运行效果如图 10.23 所示。

更新图书类的实现

图书信息

选择要更新的字段名称：○图书名称 ○作者 ○出版社 ○出版日期

输入更新内容和图书ID并用空格分割　更新

图书ID	图书名称	作者	出版社	出版日期
1	PHP范例宝典	明日科技有限公司	人民邮电出版社	2010/八月/11 0:00
2	PHP自学手册	明日科技有限公司	人民邮电出版社	2010/八月/11 0:00
3	PHP实战宝典	明日科技有限公司	人民邮电出版社	2010/八月/11 0:00

图 10.23　更新图书类的实现

关键技术

更新图书类 UpdateDemo 同样集成数据库连接类 ConnDemo，与数据库删除类的唯一不同之处是在 SQL 方法中，定义的是 update 更新语句。UpdateDemo 类的代码如下：

```
class UpdateDemo extends ConnDemo{                                                        //继承抽象类
        var $sql;                                                                         //定义变量
        public function SQL($Condition1,$Condition2,$id){                                 //定义方法拼接字符串
                $this -> sql = $sql;
                $this -> sql = "update ".$this -> tb." set $Condition1='$Condition2' where id=$id";    //拼接字符串
                return $this -> sql;
        }
        public function update(){                                                         //更新操作方法
                if($result = mssql_query($this -> sql)){
                        echo "<script>alert('更新操作以成功');location.href='index.php?id=1';</script>";    //JavaScript 提示
                }else{
                        echo "<script>alert('SQL 语句发生错误');</script>";
                }
```

```
        }
}
?>
```

设计过程

（1）在 conn.php 文件中，首先，编写 ConnDemo 数据库连接类，通过 connect()方法完成与 SQL Server 数据库的连接。本部分采用编写抽象类实现的，代码如下：

```
<?php
abstract class ConnDemo{                                            //抽象类
        var $host;                                                  //主机名称
        var $user;                                                  //用户名称
        var $pwd;                                                   //用户密码
        var $db;                                                    //数据库名称
        var $tb;                                                    //数据表名称
        var $sql;                                                   //SQL 语句
        public function __construct($host,$user,$pwd,$db,$tb){      //构造函数
                $this -> host = $host;
                $this -> user = $user;
                $this -> pwd = $pwd;
                $this -> db = $db;
                $this -> tb = $tb;
                $this -> connect();
        }
        public function connect(){                                  //连接方法
                $this -> conn = $conn;
                $this -> conn = mssql_connect($this -> host,$this -> user,$this -> pwd) or die ("Connect SQL Server false");
                mssql_select_db($this -> db,$this -> conn) or die ("Connect Database false");
        }
        public abstract function update();                          //抽象方法
}
?>
```

（2）在 conn.php 文件中，编写数据库更新类 UpdateDemo，继承自数据库连接类并重写抽象方法，定义 SQL()方法和 Update()方法实现数据的更新操作，其代码如关键技术中所示。

（3）新建 index.php 文件。首先，包含 conn.php 文件，实例化数据库更新类 UpdateDemo。然后，执行查询语句，循环输出数据表中的数据。接着，创建 form 表单，提交要更新的数据到本页。最后，根据表单中提交的数据，调用数据库更新类 UpdateDemo 中的方法执行更新操作，其关键代码如下：

```
<?php
if($_POST[sub]){                                                                        //单击按钮
        if($_POST[check] == ""){                                                        //判断单选按钮是否被选中
                echo "<script>alert('请选择要更新的字段名称');location.href='index.php?id=1';</script>";    //JavaScript 提示
        }else{
                if($_POST[te] == "输入更新内容和图书 ID 并用空格分割"){                    //判断文本框是否有内容
                        echo "<script>alert('请书写更新内容和图书 ID');location.href='index.php?id=1';</script>";    //JavaScript 提示
                }else{
                        $array = explode(" ",$_POST[te]);                               //字符串拆分
                        if(count($array) > 2 || count($array) == 1){                    //判断拆分后的数组长度
                                echo "<script>alert('文本框输入内容有误');</script>";   //JavaScript 提示
                        }else{
                                if(!preg_match("/\d/",$array[1])){                      //判断拆分后的 ID 是否是数字
                                        echo "<script>alert('输入的图书 ID 不是数字或不在范围');location.href='index.php?id=1';</script>";
                                }else{
                                        $str = $_POST[check];
                                        $str1 = iconv('utf-8','gbk',$array[0]);
                                        $id = $array[1];
                                        $ud -> SQL($str,$str1,$id);                     //调用类中方法并传递参数
                                        $ud -> update();                                //调用更新方法
                                }
                        }
                }
        }
}
?>
```

秘笈心法

心法领悟 489：抽象类。

抽象类与普通的类在形式上没有太多的差别。抽象类中必须存在至少一个抽象方法，其好处在于被其他的子类继承时可以选择性地执行其中的一个或多个抽象方法。

实例 490　通过 ADO 方式实现输入页码跳转到指定页

光盘位置：光盘\MR\10\490

高级

趣味指数：★★★★

实例说明

分页显示模块的页码跳转方式有多种，比如单击页码跳转、从下拉列表中选择页码实现跳转等。或许大多数程序员已经熟悉通过函数方式实现分页，而本例将介绍如何通过 ADO 方式操纵 SQL Server 数据库实现分页显示。熟悉 ASP 的读者可能会发现，采用这种方式实现分页与 ASP 通过记录集实现分页的方式类似。运行本实例，如图 10.24 所示，首先在图中文本框中输入要跳转的页码，然后单击“GO”按钮，即可实现页码跳转。

ADO方式实现指定页跳转

主题	发表时间	发表人	浏览次数
JAVA类相关图书	2007-12-6	who	56
JAVA类相关图书	2007-12-6	who	56
JAVA类相关图书	2007-12-6	who	56
JAVA类相关图书	2007-12-6	who	56
JAVA类相关图书	2007-12-6	who	56

共有主题406个 每页显示5个 第1页/共82页　1 2 3 4 ▶▶ ▶▶| GO

图 10.24　输入页码跳转到指定页

关键技术

为了方便操作，本实例将采用 GET 方法提交用于输入页码的表单，这样做的目的是使通过单击页码实现的跳转与通过在文本框中输入页码实现的跳转都用$_GET[]进行接收，否则还得单独做一个用$_POSI[]数组接收提交的页码的模块。通过 form 表单完成页码跳转的代码如下：

```
<form action="index.php" method="get" name="form1" id="form1">
<tr>
        <td width="30">
              <div align="center"><input type="text" name="page" size="2" class="inputcss" /></div>
        </td>
        <td width="40">
                <div align="center"><input name="submit" type="submit" class="buttoncss" value="GO" /></div>
        </td>
</tr>
</form>
```

设计过程

（1）　新建 conn.php 文件，通过 ADO 连接 SQL Server 数据库，其代码如下：

```
<?php
$conn=new com("adodb.connection");                                                          //创建记录集
$connstr="provider=sqloledb;data source=localhost;uid=sa;pwd=;database=db_database10";     //连接 SQL Server 数据库
$conn->open($connstr);                                                                      //返回连接对象
?>
```

（2）编写代码用于接收查询字符串的值（该值代表要跳转的页）并实现分页显示，代码如下：



```
                                        for($i = 1; $i <= $rs->pagecount; $i ++) {
                                                ?>
        <a href="index.php?page=<?php  echo $i;?>"><?php   echo $i;?></a>
        <?php
                                        }
                                } else {
                                        for($i = 1; $i <= 4; $i ++) {
                                                ?>
        <a href="index.php?page=<?php  echo $i;?>"><?php   echo $i;?></a>
        <?php
                                        }
                                        ?>
        <a href="index.php?page=<?php  if ($rs->pagecount >= $page + 1) echo $page + 1; else echo 1; ?>" title="后一页"><font face="webdings"> 8
</font></a>
        <a href="index.php?page=<?php  echo $rs->pagecount; ?>" title="尾页"><font face="webdings"> : </font></a>
        <?php
                                }
                                ?>
      </div>
        </td>
 </tr>
</table>
```

秘笈心法

心法领悟 490：表单提交的两种方式。

提交表单可以采取 POST 和 GET 两种方式，这两种方式的区别是：使用 POST 方法提交的信息在地址栏中不显示，并且用$_POST[]数组接收提交的内容。而采用 GET 方法提交的信息在地址栏中显示，如果提交多个信息，则信息之间用“&”连接，并且提交的内容用$_GET[]数组进行接收。

实例 491　**利用 mssql 函数库实现指定页跳转**　　高级

光盘位置：光盘\MR\10\491　　趣味指数：★★★★

实例说明

通过单击页码跳转到指定页，是众多分页方式中应用最广泛的一种，通过这种方式可以浏览某一模块中的所有信息。PHP 操作 MySQL 数据库实现分页显示主要通过关键字 limit 实现，但 limit 关键字只是 MySQL 数据库的扩展部分，不能被 SQL Server 数据库识别，所以本实例将采取一种特殊方式实现 SQL Server 数据库信息的分页展示。运行本实例，如图 10.25 所示，通过单击图中页码超链接即可实现不同页之间的跳转。

指定页跳转

主题	发表时间	发表人	浏览次数
JAVA类相关图书	2007/十二月/6 0:00	who	56
明日科技出版的VB类相关图书	2007/十二月/5 0:00	lll	90
明日科技出版的C#类相关图书	2007/十二月/4 0:00	phk	12
明日科技出版的PB类相关图书	2007/十二月/3 0:00	zts	7
明日科技出版的C++类相关图书	2007/十二月/3 0:00	lzh	4

共有主题14个 每页显示5个 第1页/共3页　　1 2 3

图 10.25　利用 mssql 函数库实现指定页跳转

关键技术

本实例采用 mssql 函数库，通过面向过程的方式实现数据的分页输出，其基本原理是：在获取数据表中总的记录数后，通过标记变量$i 来控制显示记录的范围，进而实现分页操作，其关键代码代码如下：

```
<?php
include_once ("conn.php");                                                    //包含数据库连接文件
$sql = mssql_query ( "select * from tb_demo07 order by pubtime desc", $conn );  //执行查询
$sql1 = mssql_query ( "select count(*) as total from tb_demo07", $conn );       //统计记录数
$info1 = mssql_fetch_array ( $sql1 );                                          //获取查询结果集
$total = $info1 [total];
if ($_GET [page] == "" || is_numeric ( $_GET [page] ) == false) {
        $page = 1;                                                             //定义分页变量
} else {
        $page = $_GET [page];                                                  //获取当前页码
}
$pagesize = 5;                                                                 //定义每页显示记录数
if ($total % $pagesize == 0) {
        $totalpage = $total / $pagesize;
} else {
        $totalpage = ceil ( $total / $pagesize );
}
$i = 1;
while ( $info = mssql_fetch_array ( $sql ) ) {                                 //循环输出查询结果
        if ($i > $pagesize * ($page - 1) && $i <= $pagesize * $page) {
        ?>
  <tr>
        <td height="25" bgcolor="#FFFFFF">
                <div align="left"><?php    echo iconv ( 'gbk', 'utf-8', $info [title] );?></div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
                <div align="center"><?php echo iconv ( 'gbk', 'utf-8', $info [pubtime] );?></div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
                <div align="center"><?php echo iconv ( 'gbk', 'utf-8', $info [pubuser] );?></div>
        </td>
        <td height="25" bgcolor="#FFFFFF">
                <div align="center"><?php echo iconv ( 'gbk', 'utf-8', $info [click] );  ?></div>
        </td>
</tr>
  <?php
}
$i ++;
}
?>
```

设计过程

（1）新建 conn.php 文件，通过 mssql 函数库完成与 SQL Server 数据库的连接。

（2）创建 index.php 文件，包含数据库连接文件，查询指定数据表中的数据，并且统计总的记录数，定义分页变量，通过标记变量$i 来控制显示记录的范围，其代码请参考关键技术中的内容。

（3）创建分页超链接，完成数据的分页显示，其代码如下：

```
<table width="754" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
        <td width="338" height="25">
        <div align="left">共有主题<?php echo $total;?>个 每页显示<?php  echo $pagesize;?>个 第<?php echo $page;?>页/共<?php
echo $totalpage;?>页</div>
        </td>
        <td width="346">
        <div align="right">
<?php
for($i = 1; $i <= $totalpage; $i ++) {
        ?>
  <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
  <?php
}
?>
</div>
        </td>
        <td width="70"> </td>
</tr>
</table>
```



其 frame 框架的布局如下：

```
<frameset rows="93,*,25" frameborder="0">
<frame src="pd_top.php?conn_type={$conn_type}" name="topwindow" scrolling="no">
      <frameset cols="200,*">
            <frame src="pd_left.php?conn_type={$conn_type}" name="leftwindow" frameborder="1">
            <frameset rows="37,*" frameborder="0">
                  <frame src="pd_title.php" name="titlewindow" scrolling="no">
                        <frame src="pd_main.php" name="mainwindow" scrolling="auto" frameborder="1">
            </frameset>
      </frameset>
  <frame src="pd_bottom.php" name="bottomwindow" scrolling="no" frameborder="1">
</frameset><noframes></noframes>
```

设计过程

“明日编程词典——在线体验版”网站首页文件包括模板页 index.html 和动态文件 index.php。

在 index.html 模板页中，应用框架技术设计网页的布局。

在 index.php 文件中，首先包含 Smarty 配置文件，然后创建模板变量，最后指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");                    //调用配置文件
$smarty->assign('conn_type',$_GET[conn_type]);            //设置模板变量
$smarty->display('index.html');                           //指定模板页
?>
```

秘笈心法

心法领悟 492：页面框架。

所谓框架是指网页的各部分是相互独立的页面，通过一个网页组成一个完整的网页显示在浏览器中，重复出现的内容被固定下来，每次浏览者发出对网页的请求时，只需下载发生变化的框架页面，其他的页面保持不变。

实例 493	网站模块导航设计 光盘位置：光盘\MR\10\493	高级 趣味指数：★★★★

实例说明

网站模块导航根据网站的 LOG 中传递的参数进行跳转，如果 LOG 中传递的是入门的数据，那么网站模块导航中将输出入门类中的功能模块，如图 10.27 所示。



图 10.27　入门类中功能模块展示

如果 LOG 中传递的是开发的数据，那么网站模块导航中将输出开发类中的功能模块，如图 10.28 所示。



图 10.28　开发类中功能模块展示

如果 LOG 中传递的是应用的数据，那么网站模块导航中将输出应用类中的功能模块，如图 10.29 所示。



图 10.29　应用类中功能模块展示



元件 2 的脚本代码如下：

```
on (release) {
 var type=this.loaderinfo.parameters["conn_type"];
 getURL("pd_left.php?conn_id=10121002&table_name=jszx&class_type=技术中心&conn_type=type", "leftwindow", "GET");
}
```

元件 3 中的脚本代码如下：

```
on (release) {
 var type=this.loaderinfo.parameters["conn_type"];
 getURL("pd_left.php?conn_id=10127105&table_name=ymzx&class_type=源码管理&conn_type=type", "leftwindow", "GET");
}
```

第二项技术：在超链接中应用 onMouseOver 和 onMouseOut 事件调用 JavaScript 脚本中的方法，控制鼠标滑过超链接图片时的样式。超链接的设置如下：

```
<a href="pd_left.php?conn_id=20510111&table_name=study&class_type=从零开始" target="leftwindow"><img
src="images/accidence/images_01.jpg" border="0" id="Image1" onMouseOver="MM_swapImage('Image1','','images/accidence/images_001.jpg',1)"
onMouseOut="MM_swapImgRestore()" /></a>
```

JavaScript 脚本中定义的方法如下：

```
function MM_preloadImages() { //v3.0
   var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
      var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
      if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}
function MM_swapImgRestore() { //v3.0
   var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_findObj(n, d) { //v4.01
   var p,i,x;   if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
      d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
   if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;!x&&i<d.forms.length;i++) x=d.forms[i][n];
   for(i=0;!x&&d.layers&&i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
   if(!x && d.getElementById) x=d.getElementById(n); return x;
}
function MM_swapImage() { //v3.0
   var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<(a.length-2);i+=3)
    if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
```

设计过程

网站模块导航的实现，主要应用到 pd_top.html 和 pd_top.php，以及 js 文件夹下的 flash1.js 和 top.js 文件。

（1）创建 pd_top.php 文件，其主要功能是定义模板变量传递参数以及指定模板页，其代码如下：

```
<?php
session_start();                                        //初始化 SESSION 变量
require_once 'system/system.inc.php';                   //包含配置文件、数据库连接、操作文件
$smarty->assign('conn_id',$_GET['conn_id']);            //定义模板变量传递参数
$smarty->display('pd_top.html');                        //指定模板页
?>
```

（2）创建模板页 pd_top.html。首先，通过 script 标签调用 js 脚本文件夹下的 top.js 和 flash1.js 文件。然后，定义 JavaScript 脚本，执行 flash1()方法，向 FLASH 中传递参数值。最后，根据模板变量$conn_id 的值进行判断，进而输出不同类别的功能模块，并为输出的功能模块创建超链接，链接到 pd_left.php 文件，同时传递相应的参数，其关键代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312">
<title>{$conn_type}编程词典</title>
</head>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
❶<script type="text/javascript" src="js/top.js"></script>
❷<script src="js/flash1.js"></script>
❸<body onLoad="MM_preloadImages('images/criterion/images002.jpg')">
<div style="padding-top:inherit; width:100%; height: 93px; background:url(images/images_09.jpg);">
<div style="padding-top:inherit; width:171px; height: 93px; float:left; ">
 <script type="text/javascript">
❹            flash1('{$conn_type}');
 </script>
```



如果是应用类，内容树状导航中默认展示的是“源码管理”功能模块的内容，如图 10.34 所示。



图 10.33　开发类的内容树状导航



图 10.34　应用类的内容树状导航

关键技术

在内容树状导航中，根据网站模块导航中传递的参数值，控制框架中 topwindow（网站模块导航）、leftwindow（内容树状导航）和 mainwindow（内容展示区）3 部分内容的输出，其应用到的技术如下：

（1）switch 条件控制语句。

在 pd_left.php 文件中，通过$_GET[]方法获取网站模块导航中参数 conn_id 传递的值，并将该值作为 switch 语句执行的条件，与 case 关键字指定的值进行判断，如果相同则执行相应的操作，首先，将一个模板页赋给一个模板变量。然后，通过 script 标签向 topwindow 中的 pd_top.php（网站模块导航）文件传递参数。最后，通过 script 标签向 mainwindow 中的 pd_main.php（内容展示区）文件传递参数，其关键代码如下：

```
/*
switch 语句，判断 conn_id 的值是否与 case 关键字中指定的值相同，如果相同，则执行相应的操作
*/
switch($_GET[conn_id]){
 case "20510111":                                    //指定关键字
        include 'accidence.php';                     //包含文件
        $smarty->assign('phtml','accidence.html');   //将模板文件的名称赋给模板变量
        //通过 script 脚本向指定的文件中传递参数
        echo "<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";
        echo "<script>parent.mainwindow.location.href='pd_main.php? conn_id=20510111';</script>";
        break;                                       //跳出循环
//省略了部分代码
```

（2）通过 Smarty 中的 include 函数在当前模板中包含其他模板文件。

在 switch 语句中，将指定的模板文件名称赋给一个模板变量 phtml，此时在内容树状导航的模板文件 pd_left.html 中，获取模板变量 phtml 传递的值，并且将该值作为 Smarty 中 include 函数包含的模板文件。内容树状导航模板文件 pd_left.html 的关键代码如下：

```
<script language="javascript" src="js/left.js"></script>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
<body>
<div style="width: 100%; height: 100%; padding-left:5px; padding-top:5px;">
{include file=$phtml}
</div>
</body>
```

（3）通过 Smarty 中的注册模板函数读取指定文件。

（4）通过 Smarty 中的 count_characters 和 truncate 方法统计和截取字符串。

在入门类、开发类和应用类的树状导航菜单中，都应用了 Smarty 中的 count_characters 和 truncate 方法，对指定字符串的长度进行统计和截取，进而判断数据属于几级菜单。count_characters 和 truncate 方法在本项目中应用的关键代码如下：

```
{if $subject[id12].8|count_characters=="12" && $subject[id12].8|truncate:8:""==$subject[id].8 }
```

上述代码取自 developer.html 模板文件。

设计过程

在内容树状导航概述中已经了解，内容树状导航除包含 pd_left.php 和 pd_left.html 文件外，还包含入门、开

发和应用 3 类中的 6 个文件。

（1）在 pd_left.php 文件中，获取网站模块导航超链接中传递的参数值，并且将参数值赋给 SESSION 变量，通过 switch 语句，对传递的参数值进行判断，进而控制框架中 topwindow（网站模块导航）、leftwindow（内容树状导航）和 mainwindow（内容展示区）3 部分内容的输出，其关键代码如下：

```
<?php
session_start();                                          //初始化 SESSION 变量
if($_GET[conn_id]=="" and $_GET[table_name]==""){         //判断 conn_id 和 table_name 的值为空
 $_SESSION[conn_id]="20510111";                           //设置初始值
 $_SESSION[table_name]="study";                           //设置数据库名称
 $_SESSION[class_type]="从零开始";                         //设置功能模块
}else{                                                    //如果不为空，则直接将它们的值赋给 SESSION 变量
 $_SESSION[conn_id]=$_GET[conn_id];
 $_SESSION[table_name]=$_GET[table_name];
 $_SESSION[class_type]=$_GET[class_type];
}
require 'system/system.inc.php';                          //包含数据库连接、操作方法和 Smarty 的配置方法
/*
switch 条件控制语句，判断 conn_id 的值是否与 case 关键字
中指定的值相同，如果相同，则执行相应的操作
*/
switch($_GET[conn_id]){
 case "20510111":                                         //指定关键字
        include 'accidence.php';                          //包含文件
        $smarty->assign('phtml','accidence.html');        //将模板文件的名称赋给模板变量
        //通过 script 脚本向指定的文件中传递参数
        echo "<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";
        echo "<script>parent.mainwindow.location.href='pd_main.php?conn_id= 20510111';</script>";
        break;                                            //跳出循环
 case "20610111":
        include 'accidence.php';
        $smarty->assign('phtml','accidence.html');
        echo "<script>parent.mainwindow.location.href='pd_main.php?conn_id= 26010111';</script>";
        break;
//省略了部分代码
 default:
        include 'accidence.php';
        $smarty->assign('phtml','accidence.html');
        echo "<script>parent.topwindow.location.href='pd_top.php?conn_id=20510111';</script>";            echo
"<script>try{parent.mainwindow.location.href='pd_main.php?conn_id= 20510111';}catch(e){}</script>";
        break;
}
$smarty->display('pd_left.html');                         //指定模板页
?>
```

（2）pd_left.html 模板文件中输出的内容由 switch 语句中定义的模板变量 phtml 的值决定，在 pd_left.html 文件中应用 Smarty 中的 include 函数包含由 phtml 指定的模板文件，其关键代码如下：

```
<script language="javascript" src="js/left.js"></script>
<link rel="stylesheet" type="text/css" href="css/link.css"></link>
<div style="width: 100%; height: 100%; padding-left:5px; padding-top:5px;">
{include file=$phtml}
</div>
```

（3）定义 developer.php 文件，获取 SESSION 变量中传递的值，执行查询操作，并将查询结果赋给指定的模板变量；注册模板函数，读取指定的文件，其代码如下：

```
<?php
if($_SESSION[conn_id]=="" and $_SESSION[table_name]==""){  //判断 SESSION 变量的值是否为空
 $conn_id="10127105";                                      //如果 SESSION 变量的值为空，则直接为变量赋值
 $table_name="ymzx";
 $class_type="源码管理";
}else{                                                     //如 SESSION 变量不为空，将 SESSION 的值赋给指定的变量
 $conn_id=$_SESSION[conn_id];
 $table_name=$_SESSION[table_name];
 $class_type=$_SESSION[class_type];
}
 $smarty->assign("conn_id",$conn_id);                      //设置模板变量
```





10.4 SQL Server 数据库的安全与维护

实例 496 **压缩数据库**

光盘位置：光盘\MR\10\496

高级

趣味指数：★★★★

实例说明

数据库在使用一段时间后，时常会出现因删除数据而造成的空闲空间太多的情况，这时就需要减少分配给数据库文件和事务日志文件的空间，以免浪费磁盘空间。当数据库中没有数据时，可以直接修改数据库文件的属性以改变其占用空间，但当数据库中有数据时，这样操作就会破坏其中的数据，因此需要使用收缩的方式来缩减数据库空间。本实例实现将 SQL Server 数据库 model 的未使用空间缩为数据库大小的 10%。在查询分析器中的运行结果如图 10.36 所示。

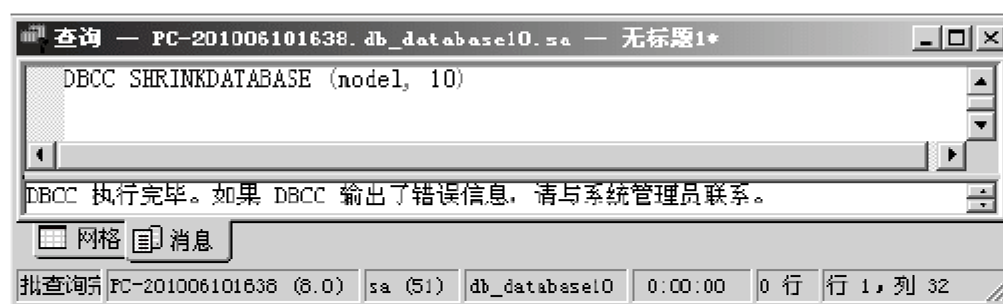


图 10.36　收缩数据和日志

关键技术

可以使用 DBCC SHRINKDATABASE 命令来压缩数据库。

DBCC SHRINKDATABASE 语句的语法如下：

```
DBCC SHRINKDATABASE
    ( database_name [ , target_percent ]
        [ , { NOTRUNCATE | TRUNCATEONLY } ]
    )
```

参数说明：

database_name：要压缩的数据库名称。

target_percent：数据库压缩后，数据库文件中所要的剩余可用空间百分比。如果指定的百分比过大，超过了压缩前未使用的空间所占的比例，则数据库不会被压缩。另外，压缩后的数据库不能比数据库初使设定的容量小。

NOTRUNCATE：在数据库文件中保留所释放的文件空间。如果未指定，将释放的文件空间分配给操作系统。

TRUNCATEONLY：将数据库文件中的任何未使用的空间释放给操作系统，并将文件收缩到上一次所分配的大小，从而减小文件，而不移动任何数据，不试图重新定位未分配页的行。选择此项后，target_percentis 选项将无效。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”，登录名为 sa，密码为空。

（2）在查询分析器中选择要连接的数据库为 db_database10。

（3）在代码编辑区中输入 SQL 语句：

```
DBCC SHRINKDATABASE (model, 10)
```

（4）单击执行按钮▶，执行本实例，即可将数据库压缩。

秘笈心法

心法领悟 496：压缩数据库。

数据库的负载越大，其运行速度就越慢。读者以后的就业取向可能是 PHP 程序员，而对于数据库的优化，取得高效的运行速度应该是一位合格的程序员所必备的技术。压缩数据库就是如何优化数据库速度的第一课。

实例 497　收缩日志和数据　高级

光盘位置：光盘\MR\10\497　趣味指数：★★★★

实例说明

本实例实现的是将 master 系统数据库中标识（ID）号为 1 的文件收缩到 7MB，运行效果图如图 10.37 所示。



图 10.37　收缩数据和日志

关键技术

使用 DBCC SHRINKFILE 命令来收缩相关数据库的指定数据文件或日志文件大小，语法如下：

```
DBCC SHRINKFILE
    ( { file_name | file_id }
        { [ , target_size ]
            | [ , { EMPTYFILE | NOTRUNCATE | TRUNCATEONLY } ]
        }
    )
```

参数说明：

file_name：已收缩文件的逻辑名称。

file_id：要收缩文件的标识(ID)号。若要获得文件的 ID，请使用 FILE_ID 函数或在当前数据库中搜索 sysfiles。

target_size：指定文件收缩后的大小，以 MB 为单位。如果不指定此选项，SQL Server 将最大限度地缩减文件。

EMPTYFILE：指明文件不再使用，将所有此文件中的数据移动到同一文件组中的其他文件中去。执行此命令后，此文件就可以用 ALTER DATABASE 语句删除了。

参数 NOTRUNCATE 和 TRUNCATEONLY 与 DBCC SHRINKDATABASE 命令中的含义相同，请读者参见实例 496，这里不再赘述。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
USE master
SELECT FILE_ID('master')
DBCC SHRINKFILE (1, 7)
```

（3）单击执行按钮▶，执行本实例，即可实现数据的收缩。

秘笈心法

心法领悟 497：收缩数据与压缩数据。

在笔者看来，收缩数据与压缩数据，在功能上没有什么不同。压缩数据是按比例将数据的物理内存减小，而收缩数据是根据传入参数将数据文件的大小减少至一定值。

实例 498　备份数据库　高级

光盘位置：光盘\MR\10\498　趣味指数：★★★★

实例说明

在网站开发过程中，数据库一般都保存有许多重要的数据信息，这些数据的丢失将会带来很严重的后果，所以对系统数据库进行备份非常重要，这样便可以在数据库出现错误时能够及时地恢复数据库。本实例介绍使用 SQL 语句备份 SQL Server 数据库的过程，并让 SQL Server 在完成备份的 20%时显示备份进度信息。

关键技术

本实例使用 BACKUP DATABASE 语句，完成数据库、差异式或文件和文件组的备份，其简要用法如下：

```
BACKUP DATABASE { database_name | @database_name_var }
TO < backup_device > [ ,...n ]
[ WITH
[ BLOCKSIZE = { blocksize | @blocksize_variable } ]
      [ [ , ] NAME = { backup_set_name | @backup_set_name_var } ]
      [ [ , ] DESCRIPTION = { 'text' | @text_variable } ]
      [ [ , ] DIFFERENTIAL ]
      [ [ , ] FORMAT | NOFORMAT ]
      [ [ , ] { NOUNLOAD | UNLOAD } ]
      [ [ , ] RESTART ]
      [ [ , ] STATS [ = percentage ] ]
]
```

参数说明如表 10.2 所示。

表 10.2　BACKUP DATABASE 语句的参数说明

参　　数	说　　明
BLOCKSIZE	设置写入时的数据区块大小。此值通常不需设置，可以由 SQL Server 自行决定，但若要将备份数据库刻录到光盘上，并打算用光盘来进行还原，则需将此值设置为 2048。若要指定磁盘的数据区块大小，必须搭配 FORMAT 参数使用
NAME	表示备份结果集的名称，其长度不超过 128 个字符
DESCRIPTION	描述备份设置的文本，最多包括 255 个字符
DIFFERENTIAL	表示仅备份上次进行完整数据库备份以来，数据所发生的变化，即进行差异备份。差异备份一般比完整备份占用更少的空间。对于上一次完整备份时备份的全部单个日志，使用该选项可以不必再进行备份
FORMAT|NOFORMAT	设置是否先格式化磁盘
NOUNLOAD|UNLOAD	设置是否要在备份完成时退出磁盘
RESTART	指定 SQL Server 重新启动一个被中断的备份操作。因为 RESTART 选项在备份操作中断处重新启动该操作，所以节省了时间。若要重新启动一个特定的备份操作，要重复整个 BACKUP 语句并且加入 RESTART 选项。不一定非要使用 RESTART 选项，但它具有节省时间的优势
STATS	表示在备份完某一百分比数据之后，所要显示的信息

设计过程

下面将备份 db_database10 数据库，具体操作步骤如下：

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
BACKUP DATABASE db_sql
TO disk='E:\MyData\db_sql 备份.bak'
```

（3）单击执行按钮，执行备份。备份后的 db_database10 数据库被保存在 E 盘的 SQL Server Data 文件夹中，如图 10.38 所示。



图 10.38　备份数据库文件

秘笈心法

心法领悟 498：备份数据库的注意事项。

笔者在讲解 MySQL 数据库备份时已经说过，备份出来的文件虽然已经经过加密处理，但是也是将数据信息以近乎明文的方式展现给用户，存在一定的风险。

实例 499　备份事务日志　高级

光盘位置：光盘\MR\10\499　趣味指数：★★★★

实例说明

本实例实现的是在创建的命名备份设备上创建 db_database10 数据库的事务日志备份。事务日志备份仅制作日志文件的复本。日志文件备份本身不能用于还原数据库。日志文件在数据库还原后，可将数据库恢复到原始的故障点。

关键技术

本实例应用的技术有创建备份设备和应用 BACKUP BLOG 语句备份事务日志。

1. 创建备份设备

备份设备是创建备份和恢复数据库的前提条件，可以将数据库、事务日志和文件备份到磁盘和磁带上。设备可以分为 4 种：磁盘设备、磁带设备、物理设备和逻辑设备。

磁盘设备、磁带设备和物理设备在备份时，只要选择相应的设备即可。而逻辑设备用来标识物理设备的别名或公用名称，其名称永久地存储在 SQL Server 内的系统表中。使用逻辑备份的优点是引用名称简单，例如，逻辑设备名称可以是 mrsoft_Backupfdw，而物理设备名称则是 E:\SQL Server Data\mrosft\MyBackup.bak。

在 Transact-SQL 语句中，使用 sp_addumpdevice 系统存储过程创建备份设备，语法如下：

```
sp_addumpdevice [ @devtype = ] 'device_type' ,
      [ @logicalname = ] 'logical_name' ,
      [ @physicalname = ] 'physical_name'
      [ , { [ @cntrltype = ] controller_type
                | [ @devstatus = ] 'device_status'
            }
      ]
```

参数说明：

device_type：备份设备的类型。device_type 的数据类型为 varchar(20)，没有默认设置，可以是 Disk（硬盘文件）、Pipe（命名管理）或 Tape（磁带设备）。

Logicalname：备份设备的逻辑名称，该逻辑名称用于 BACKUP 和 RESTORE 语句中，其数据类型为 sysname，没有默认值，并且不能为 NULL。

Physicalname：备份设备的物理名称。物理名称必须遵照操作系统文件名称的规则或者网络设备的通用命名规则，并且必须包括完整的路径。physical_name 的数据类型为 nvarchar(260)，没有默认值，并且不能为 NULL。

2. 用 BACKUP LOG 语句备份事务日志

执行 BACKUP LOG 语句备份事务日志，同时要指定要备份的事务日志所属的数据库名称、事务日志备份将写入的备份设备。其语法格式请参见实例 498 中的技术分析。

执行完整数据库备份或差异数据备份期间不能备份事务日志，但在运行文件备份的同时可以备份事务日志。另外，下列情况不要备份事务日志：

（1）在由于事务日志包含自上次创建备份后数据库所发生的更改，而创建了数据库或文件备份之前。

（2）事务日志已被截断。此时如果要备份事务日志，必须先创建数据或差异数据库备份。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句，创建一个名为 mrsoft_Backupfdw 的逻辑磁盘设备，其物理路径为 E:\SQL Server Data\mrosft\MyBackup.bak。

```
SP_ADDUMPDEVICE 'disk','mrsoft_Backupfdw',' E:\SQL Server Data\mrosft\MyBackup.bak '
```

（3）单击执行按钮▶，执行本实例，其运行结果如图 10.39 所示。

（4）在代码编辑区中输入如下 SQL 语句，用来在创建的命名备份设备 mrsoft_Backupfdw 上创建 db_database10 数据库的事务日志备份。

```
BACKUP LOG db_database10
TO mrsoft_Backupfdw
GO
```

（5）单击执行按钮▶，执行本实例，其运行结果如图 10.40 所示。

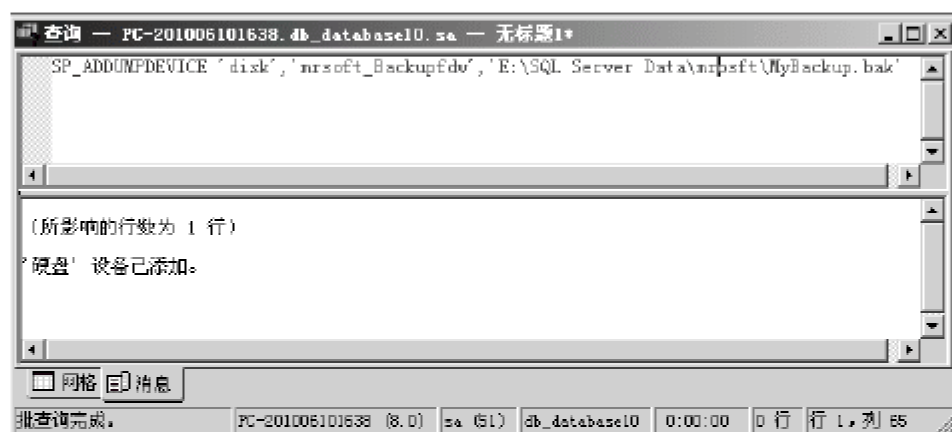


图 10.39　创建备份设备

BACKUP LOG 操作成功地处理了 0 页，花费了 0.027 秒（0.000 MB/秒）。

图 10.40　事务日志备份

秘笈心法

心法领悟 499：SQL Server 备份事务日志的注意事项。

在 SQL Server 中备份 db_database10 数据库事务日志的过程中，首先选择数据库的故障还原模式，如选择“完全”模式，然后单击“确定”按钮，如图 10.41 所示。

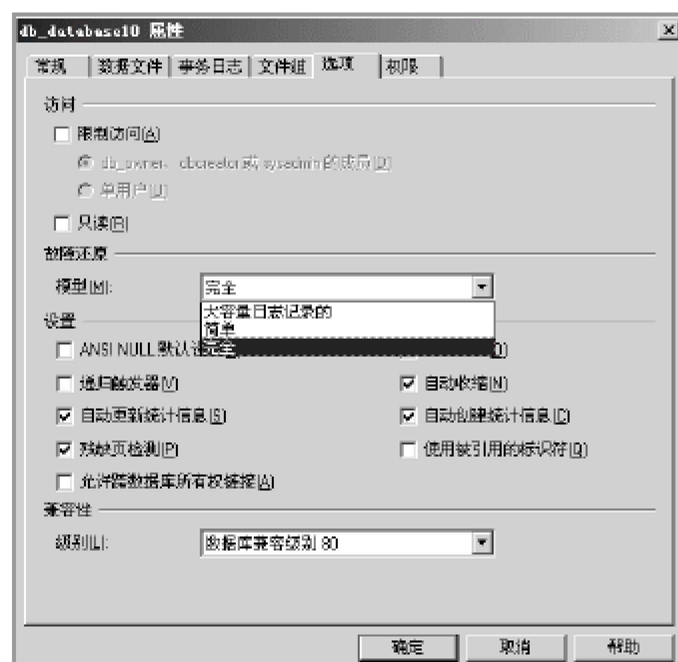


图 10.41　db_database10 数据库故障还原模式



读，但不尝试验证备份卷中的数据结构。如果备份有效，则在 SQL Server 2000 中返回此消息："备份集有效"。

实例 501	定时备份数据库 光盘位置：光盘\MR\10\501	高级 趣味指数：★★★★

实例说明

本实例实现定时备份数据库，并让 SQL Server 在备份完成 20%时显示备份进度信息。在查询分析器中的运行结果如图 10.43 所示。

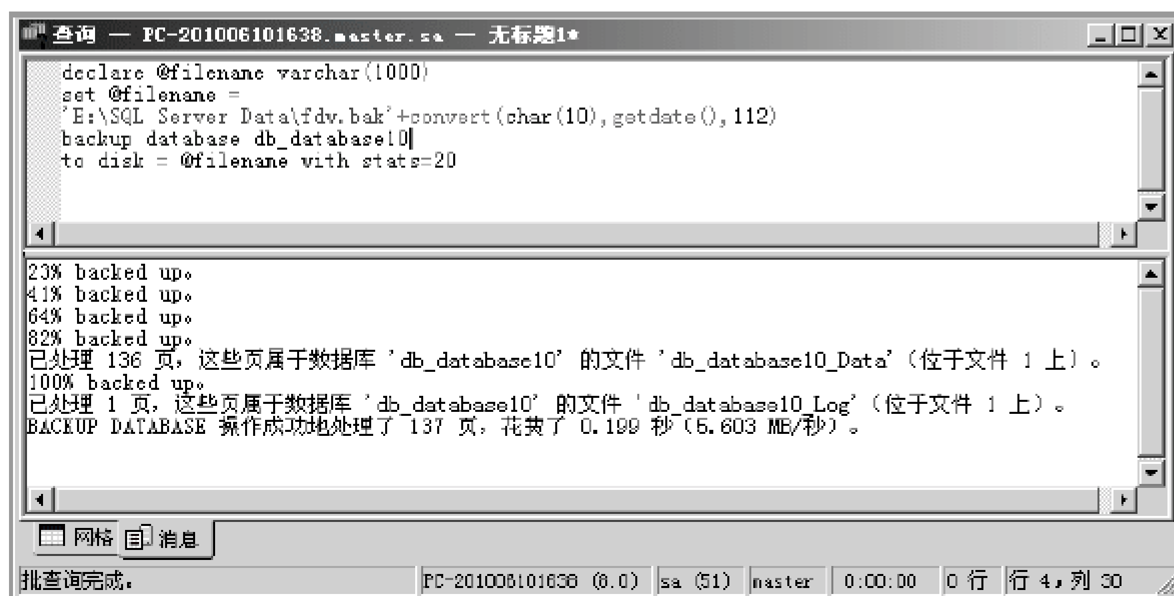


图 10.43　定时备份数据库

关键技术

本实例主要应用转换函数 CONVERT 实现。在将 datetime 数据转换为 char 或 varchar 数据时，CONVERT 的 style 参数提供了各种日期显示格式。年份可以显示为两位或四位数。默认情况下，SQL Server 将年份显示为两位数，若要显示包括世纪的四位数年份（即使年份数据是使用两位数的年份格式存储的），给 style 值加 100 即可。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
declare @filename varchar(1000)
set @filename =
'E:\SQL Server Data\fdw.bak'+convert(char(10),getdate(),112)
backup database db_database10
to disk = @filename with stats=20
```

（3）单击执行按钮▶，执行本实例，其实现过程如图 10.44 所示。

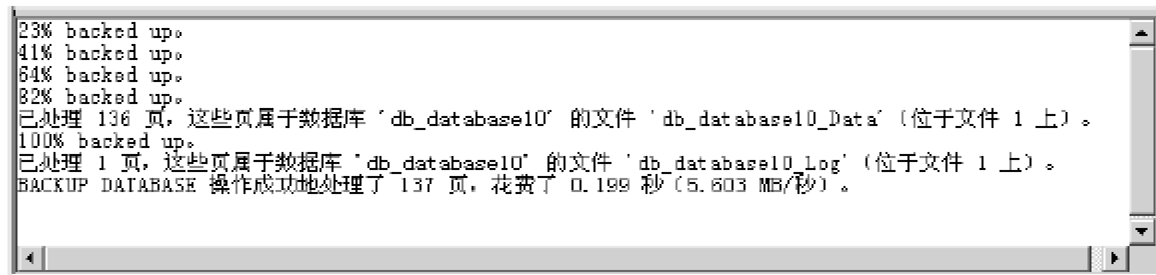


图 10.44　本实例的实现过程

秘笈心法

心法领悟 501：用 style 参数转换日期。

用 style 参数转换日期的语句如下：

```
SELECT CONVERT(char(12),GETDATE(),3)
```

此语句把当前日期转换为样式 3，即 dd/mm/yy。

实例 502　数据库中数据的加密

光盘位置：光盘\MR\10\502

高级

趣味指数：★★★★

实例说明

本实例实现对 db_database10 数据库 tb_demo11 用户信息表中的用户名和密码进行加密。注意，在通过 pwdencrypt()函数对相同的用户名和密码进行加密后，生成的结果却是不相同并且是不可逆的。本实例的运行结果如图 10.45 所示。

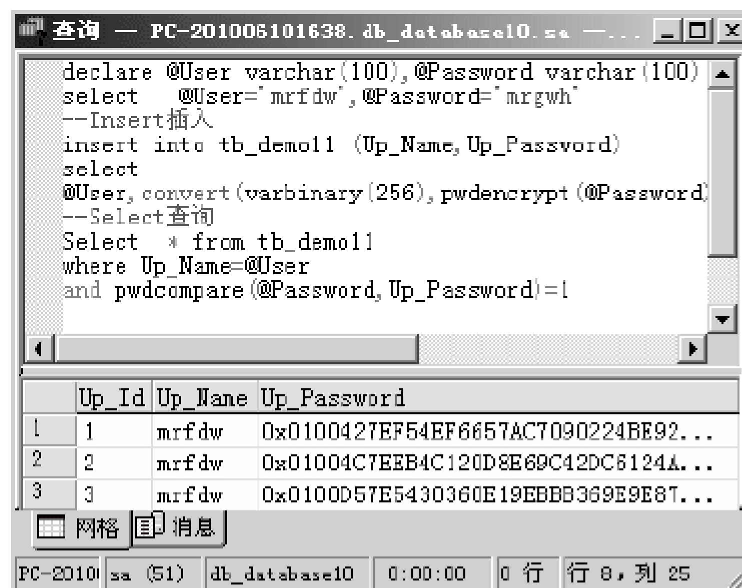


图 10.45　对数据库中数据进行加密

关键技术

本实例主要应用加密函数 pwdencrypt()和加密校验函数 pwdcompare()实现。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
declare @User varchar(100),@Password varchar(100)
select     @User='mrfdw',@Password='mrgwh'
--Insert 插入
insert into dbo.tb_demo11 (Up_Name,Up_Password)
select
@User,convert(varbinary(256),pwdencrypt(@Password))
--Select 查询
Select   * from tb_demo11
where Up_Name=@User
and pwdcompare(@Password,Up_Password)=1
```

（3）单击执行按钮▶，执行本实例，其实现的结果如图 10.45 所示。

秘笈心法

心法领悟 502：加密函数 pwdencrypt()。

由于此函数对插入的相同数据进行加密后，生成的结果不相同而且不可逆，是一种非常安全的加密手段，被广泛应用。

实例 503　SQL Server 数据库的分离和附加

光盘位置：光盘\MR\10\503

高级

趣味指数：★★★★

实例说明

在编写本章目录时，本实例的内容并未纳入编写计划之中。但是笔者认为，每位用户对 SQL Server 数据库的掌握程度不尽相同，为了方便读者运行和学习本章实例内容，避免出现不必要的操作失误，特向读者介绍一些操作 SQL Server 数据库的细节内容。本实例主要讲解如何分离、附加数据库。

关键技术

分离数据库：将数据库从 SQL Server 数据库中分离出来以复制数据库文件。已分离的数据库将无法使用，除非再次将其附加到 SQL Server 中。

附加数据库：是相对于分离数据库而言的，当数据库文件被分离后，重新载入到数据库中。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“企业管理器”命令，启动企业管理器。

（2）选中指定的数据库，单击鼠标右键，选择“所有任务”/“分离数据库”命令，如图 10.46 所示。

（3）附加数据库。选中“数据库”选项，单击鼠标右键，选择“所有任务”/“附加数据库”命令，如图 10.47 所示。选择指定的数据库文件，单击“确定”按钮完成附加。

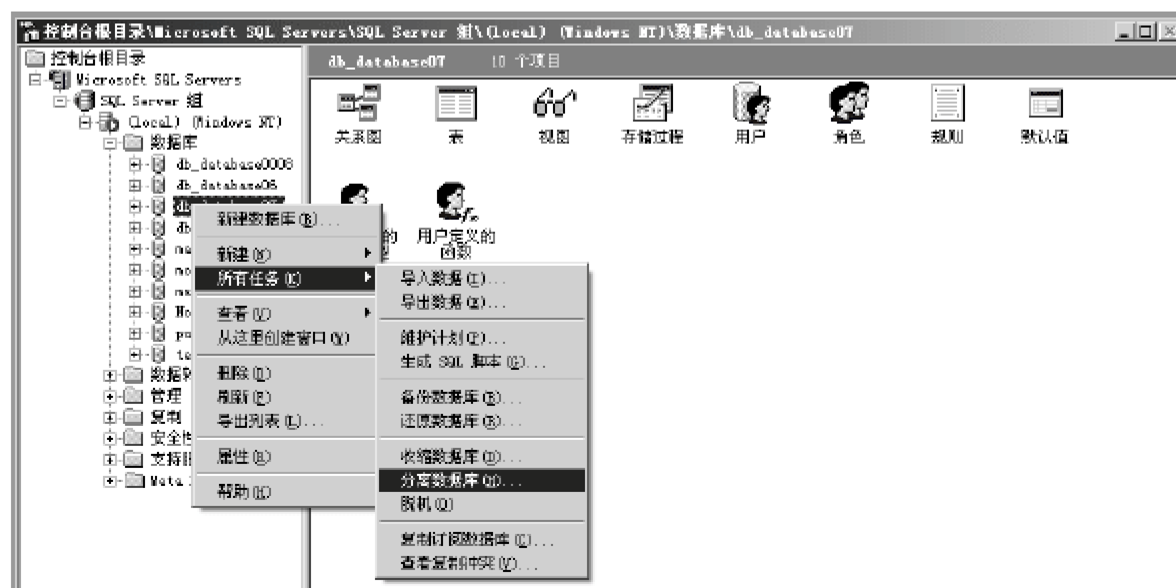


图 10.46　分离数据库



图 10.47　附加数据库

秘笈心法

心法领悟 503：SQL Server 数据库的附加。

如果要将其他计算机上的数据库文件转到用户机上，可以在 SQL Server 安装路径的 Date 文件夹下，复制或剪切该数据库文件，然后将其粘贴到用户机 SQL Server 安装路径的 Date 文件夹中，随即附加数据库即可。

实例 504　从 SQL Server 数据库导出到 Excel 表中　高级

光盘位置：光盘\MR\10\504　趣味指数：★★★★

实例说明

Excel 是集数据统计、数据分析、图表制作、打印等功能于一身的专业电子表格软件，可以将数据导出到 Excel 中进行各种处理。本实例讲解将 SQL Server 中 db_database10 数据库中的数据表 tb_demo10 的数据导出到 Excel（tb_demo10.xls）文件中的方法，运行结果如图 10.48 所示。

Microsoft Excel - tb_demo10

	A	B	C	D
1	1	《PHP开发实战宝	78元	########
2	2	《PHP标准教程》	79元	########
3	3	《PHP网络编程宝	73元	########
4	4	《PHP范例宝典》	98元	########
5	5	《PHP范例手册》	68元	########

图 10.48　Excel（tb_demo10.xls）文件中的数据

关键技术

本实例主要应用 xp_cmdshell 函数和 bcp 实用工具实现功能。

（1）xp_cmdshell 函数。

xp_cmdshell：以操作系统命令行解释器的方式执行给定的命令字符串，并以文本行方式返回任何输出，其语法如下：

```
xp_cmdshell {'command_string'} [, no_output]
```

参数说明：

'command_string'：在操作系统命令行解释器上执行的命令字符串。command_string 的数据类型为 varchar(255) 或 nvarchar(4000)，没有默认值。command_string 不能包含一对以上的双引号，如果所引用的文件路径或程序名称中有空格，则需要使用一对引号。

no_output：可选参数，表示执行给定的 command_string，但不向客户端返回任何输出。

返回代码值：0（成功）或 1（失败）。

（2）bcp 实用工具

bcp 实用工具是一种命令提示符大容量复制的实用工具，它以用户指定的格式从操作系统文件或向操作系统文件复制。下面以 bcp 实用工具用于 SQL Server 2000 实例和数据文件之间指定的格式复制数据为例，来介绍 bcp 实用工具中几个重要的参数。

- ☑ In|out|queryout|format：指定大容量复制的方向。In 是指从文件复制到数据库表或视图，out 是指从数据库表或视图复制到文件。
- ☑ -c：使用字符数据类型执行大容量复制操作。
- ☑ -q：在 bcp 实用工具和 SQL Server 实例的连接中执行 SET QUOTED_IDENTIFLERSON 语句。
- ☑ -S server_name[/instance_name]：指定要连接的 SQL Server 实例。
- ☑ -U login_id：指定用于连接的 SQL Server 实例。
- ☑ -P password：指定登录 ID 的密码。

设计过程

（1）选择“开始”/“所有程序”/Microsoft SQL Server/“查询分析器”命令，在弹出的连接对话框中选择“SQL Server 身份验证”选项，登录名为 sa，密码为空。

（2）在代码编辑区中输入 SQL 语句：

```
EXEC master..xp_cmdshell 'bcp db_database10.dbo.tb_demo10 out E:\tb_demo10.xls -c -q -S"localhost"
-U"sa" -P""'
```

（3）单击执行按钮▶，执行本实例，其运行结果如图 10.49 所示。

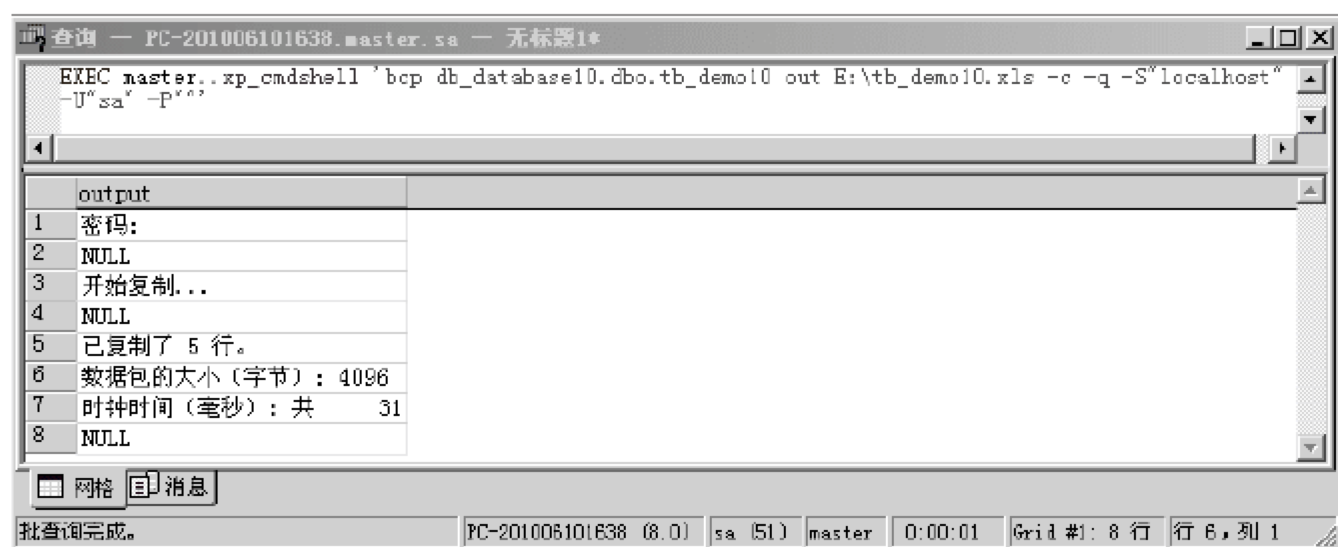


图 10.49　从 SQL Server 数据库导出到 Excel 表中

秘笈心法

心法领悟 504：使用 xp_cmdshell 函数的注意事项。



第11章

PHP操作Access数据库

- 连接数据库
- 操作 Access 数据库
- Access 数据库的安全
- Access 数据库的综合运用

11.1　连接数据库

如果开发一个小型网站，利用 Access 数据库不仅方便操作、易于维护，更重要的是可以节省服务器资源，下面将通过具体实例讲解 PHP 操纵 Access 数据库的方法。

实例 506　通过 COM 类连接 Access 数据库

光盘位置：光盘\MR\11\506

高级

趣味指数：★★★★

实例说明

本实例主要介绍 PHP 如何通过 ADO 方式连接 Access 数据库，运行本实例如图 11.1 所示，该页面显示的信息是从 Access 数据库中检索出来的，从而证明通过 ADO 方式连接数据库成功。

通过COM类连接ACCESS数据库

书名	出版社
《PHP数据库开发完全手册》	人民邮电出版社
《PHP范例宝典》	人民邮电出版社

图 11.1　连接 Access 数据库

关键技术

ADO 技术基于通用对象模型（COM），可提供多种语言的访问技术。PHP 是通过预先定义类 com 来使用 ADO 方法操纵 Access 数据库的，该类的详细说明如下：

```
string com::com( string module_name [, string server_name [, int codepage]])
```

参数说明：

module_name：被请求组件的名字或 class-id。

server_name：DCOM 服务器的名字。

Codepage：指定用于将 PHP 字符串转换成 UNICODE 字符串的代码页，反之亦然。该参数的取值有 CP_ACP、CP_MACCP、CP_OEMCP、CP_SYMBOL、CP_THREAD_ACP、CP_UTF7 和 CP_UTF8。

设计过程

（1）建立数据库连接文件 conn.php，实现与 Access 数据库的连接，代码如下：

```
<?php
	$conn = new com("adodb.connection");
	$connstr="driver={microsoft access driver (*.mdb)}; dbq=". realpath("data/db_database11.mdb");
	$conn->open($connstr);
?>
```

（2）建立如图 11.1 所示的页面，并循环显示数据库中的所有图书信息，代码如下：

```
 <?php
   $sql="select * from tb_book";
   $rs=new com("adodb.recordset");
   $rs->open($sql,$conn,1,3);
   while(!$rs->eof)
   {
  ?>
       <tr>
         <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo iconv('gbk','utf-8',$rs->fields(bookname)->value);?></div></td>
         <td height="25" bgcolor="#FFFFFF"><div align="center"><?php echo iconv('gbk','utf-8',$rs->fields(pub)->value);?></div></td>
       </tr>
```

```
  <?php
      $rs->movenext;
    }
?>
```

秘笈心法

心法领悟 506：ADO 的相关了解。

ADO 向我们提供了一个熟悉的、高层的对 OLE DB 的 Automation 封装接口，对那些熟悉 RDB 的程序员来说，可以把 OLE DB 比作 ODBC 的驱动程序。目前，虽然 OLE DB 的提供者较少，但微软正积极更新该技术并打算用 OLE DB 取代 ODBC。

实例 507　连接带密码的 Access 数据库

光盘位置：光盘\MR\11\507

高级

趣味指数：★★★★

实例说明

虽然 Access 是开发小型项目的首选数据库，但是，商家不会因为项目小而不要求其安全性，所以给 Access 数据库设置密码是十分必要的。本实例通过 ADO 方式连接带密码的 Access 数据库。当密码错误时，将显示如图 11.2 所示的错误信息。

```
Fatal error: Uncaught exception 'com_exception' with message '<b>Source:</b> Microsoft OLE DB Provider
for ODBC Drivers<br/><b>Description:</b> [Microsoft][ODBC Microsoft Access Driver] 密码无效。' in
F:\xampp\xampp\htdocs\MR\11\002\conn.php:4 Stack trace: #0 F:\xampp\xampp\htdocs\MR\11\002\conn.php(4):
com->open('driver={microso...') #1 {main} thrown in F:\xampp\xampp\htdocs\MR\11\002\conn.php on
line 4
```

图 11.2　连接 Access 数据库密码错误

关键技术

本实例实现连接带密码的 Access 数据库的代码如下：

```
<?php
    $conn = new com("adodb.connection");
    $connstr="driver={microsoft access driver (*.mdb)};Pwd=111; dbq=". realpath("data/db_database11.mdb");
    $conn->open($connstr);
?>
```

其关键是在 ADO 中设置 Pwd 的参数值，即 Access 数据库的密码。

设计过程

（1）设置 Access 数据库的密码与操作其他数据库不同，必须以独占方式打开数据库。

（2）选择"工具"/"安全"/"设置数据库密码"命令，在打开的对话框中设置密码和确认密码，如图 11.3 所示。

（3）建立网页文件，输入如关键技术中所示的代码。如没有任何输出，表示连接 Access 成功。

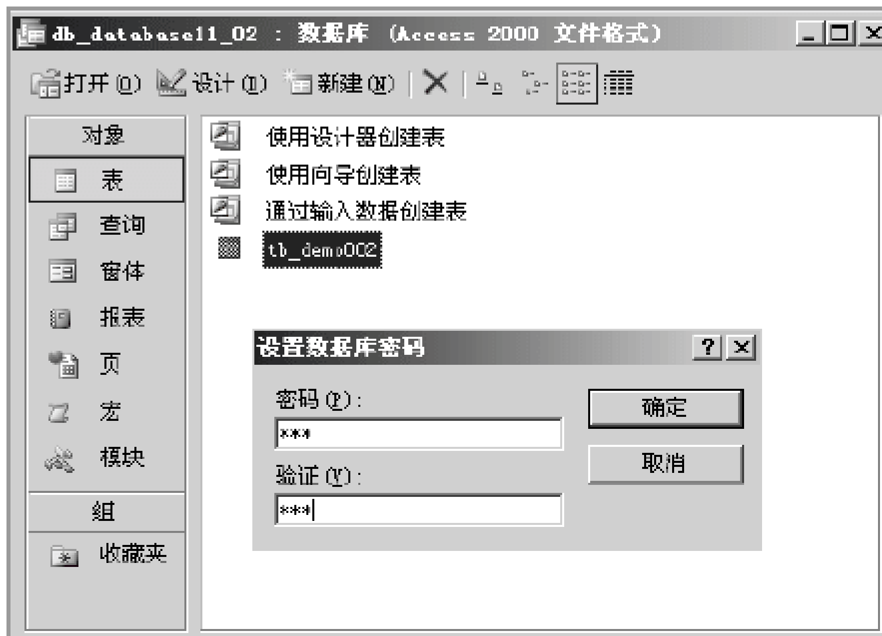


图 11.3　设置并确认密码

秘笈心法

心法领悟 507：取消或更改 Access 数据库的密码。

当用户对 Access 数据库进行维护时，为了方便起见，可以暂时撤销 Access 数据库的密码，方法是：以独占方式打开数据库，输入密码并进入数据库；选择"工具"/"安全"/"撤销数据库密码"命令，根据提示再次输

入密码，单击“确定”按钮即可。

11.2 操作 Access 数据库

Access 数据库是小型项目后台支持的首选，在相关的数据库操作上与其他数据库有所不同。本节将通过实例详细介绍如何操作 Access 数据库。

实例 508　通用数据信息的查询　　高级

光盘位置：光盘\MR\11\508　　趣味指数：★★★★

实例说明

本实例主要实现 Access 数据库信息的查询。运行本实例，在文本框中输入某学生的学号，如果查询到该学生的成绩将该学生的详细成绩信息显示出来，如图 11.4 所示。

成绩查询

请输入学号：0312101 查询

学号	姓名	班级	科目	成绩
0312101	小刘	三年一班	语文	10
0312101	小刘	三年二班	代数	22

不及格科目：2门

图 11.4　通用数据信息查询

关键技术

通过 ADO 方式建立与 Access 数据库的连接已经在前面介绍过，但建立完该连接后并不能对数据库进行查询、更新等操作，必须先实例化 com 类，建立记录集，然后通过 open()方法执行查询语句，其关键代码如下：

```
include("conn.php");                                              //包含数据库连接文件
$rs=new com("adodb.recordset");                                   //建立记录集
$rs->open("select * from tb_score where sno='".$_POST[sno]."'",$conn,3,1);   //执行查询
```

设计过程

（1）新建 conn.php 文件，建立与 Access 数据库的连接，代码如下：

```
<?php
$conn=new com("adodb.connection");
$connstr="provider=microsoft.jet.oledb.4.0;data source=".realpath("../data/db_database11.mdb");
$conn->open($connstr);
?>
```

（2）新建 index.php 文件，创建 form 表单，将学号提交到本页，在本页中根据 POST 方法提交的学号执行查询操作，查询出符合条件的数据，并且输出查询结果，代码如下：

```
<?php
   if($_POST[submit]!=""){
        include("conn.php");
          $rs=new com("adodb.recordset");
          $rs->open("select * from tb_score where sno='".$_POST[sno]."'",$conn,3,1);
          if($rs->eof|| $rs->bof){
          ?>
         <tr>
            <td height="20" colspan="5" bgcolor="#FFFFFF"><div align="center">没有查找到该学生成绩！</div> </td>
            </tr>
          <?php
```



实例 509　查询结果的分页显示

光盘位置：光盘\MR\11\509

高级

趣味指数：★★★★

实例说明

如果记录集中的记录较多，用一个页面显示所有记录会给用户的浏览带来诸多不便。为了解决这个问题，可以对数据库中的数据进行分页显示。运行本实例，如图 11.5 所示。



图 11.5　数据的分页显示

关键技术

PHP 利用 COM 类连接数据库，可以直接使用类中的数据成员完成很多操作，常用的数据成员如表 11.3 所示。

表 11.3　记录集对象常用数据成员

数 据 成 员	说　　明
pagesize	设置分页显示时，记录集对象一页所容纳的记录数
absolutepage	设置当前显示的为第几页，首先应设置 pagesize 的值
bof	检查当前记录集对象所指位置是否为第一条记录之前
eof	检查当前记录集对象所指位置是否为最后一条记录之后
pagecount	显示记录集当前页面总数，首先应设置 pageesize 的值
movenext	将记录集对象的指针向下移动一条

设计过程

（1）建立数据库连接文件 conn.php，代码如下：

```
<?php
	$conn = new com("adodb.connection");
	$connstr="driver={microsoft access driver (*.mdb)}; dbq=".
	realpath("../data/db_database11.mdb ");
	$conn->open($connstr);
?>
```

（2）根据接收到的查询字符串的值，确定显示的信息页，代码如下：

```
<?php
	include("Conn/conn.php");                                  //包含数据库连接文件 conn.php
	$sql="select * from tb_bookinfo order by pdate desc";      //查询 tb_bookinfo 表中的所有商品
	$rs=new com("adodb.recordset");                            //用 com 类声明一个记录集对象
	$rs->open($sql,$conn,1,3);                                 //利用 com 类的 open()方法执行查询
	$rs->pagesize=20;                                          //设置每页最多显示 20 条记录
```

```
    if((trim(intval($_GET[page]))=="")||(intval($_GET[page])>$rs->pagecount)||(intval($_GET[page])<=0)) {
            $page=1;
    } else {                          //否则使$page 的值为所接收的查询字符串的数值
            $page=intval($_GET[page]);
    }
    if($rs->eof || $rs->bof) {        //判断表中的内容是否为空，如果为空则给出提示
?>
    <tr>
    <td height="20" colspan="5" bgcolor="#FFFFFF"><div align="center">本站暂无商品！
    </div></td>
    </tr>
    <?php
    }else{                            //不为空则分页显示数据
          $rs->absolutepage=$page;    //设置当前显示页为所接收的查询字符串的值
          $mypagesize=$rs->pagesize;  //定义变量$mypagesize 用于控制当前页的循环终止
       while(!$rs->eof && $mypagesize>0) {  //记录集对象没有到整个表尾并且当前页所要显示的记录没有完全显示完，则循环显示数据信息
 ?>
……
<?php
       $mypagesize--;                 //每显示一条记录，应使$mypagesize 的值减 1
       $rs->movenext;                 //使记录集指针移到下一条记录的位置
      }
}
?>
```

（3）向信息显示页面发送查询字符串从而实现商品信息的分页显示，代码如下：

```
        <div align="right">
<?php
    if($page>=2){
?>
   <a href="index.php?page=1" title="首页"><font face="webdings"> 9 </font></a>
   <a href="index.php?page=<?php echo $page-1;?>" title="前一页"><font face="webdings"> 7            </font></a>
<?php
      }
      if($rs->pagecount<=4){
             for($i=1;$i<=$rs->pagecount;$i++){
?>
               <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
<?php
             }
       }else{
        for($i=1;$i<=4;$i++){
?>
                <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
<?php
        }
?>
          <a href="index.php?page=<?php
         if($rs->pagecount>=$page+1)
              echo $page+1;
          else
              echo 1;
        ?>" title="后一页"><font face="webdings"> 8 </font></a>
       <a href="index.php?page=<?php echo $rs->pagecount;?>" title="尾页"><font face="webdings"> : </font></a>
       <?php
            }
        ?>
</div>
```

秘笈心法

心法领悟 509：比较 ASP 与 PHP。

熟悉 ASP 的读者可能会发现，PHP 利用 COM 类连接 Access 数据库的思路与 ASP 用 ADO 方法连接数据库的思路类似，同样实现翻页的思路也是类似的。



```
		<td height="25" bgcolor="#FFFFFF"><div align="left"> <input type="password" name="userpwd" size="22"
class="inputcss"></div></td>
	</tr>
  <tr>
		<td height="25" colspan="2" bgcolor="#FFFFFF"><div align="center"><input name="submit" type="submit" value="登录"
class="buttoncss"></div></td>
		</tr>
</form>
```

（2）建立密码验证类。

（3）类的实例化，并实现用户身份的验证，代码如下：

```
$chk1=new chk($username,$userpwd);
$chk1->chkuser();
```

秘笈心法

心法领悟 510：类的实例化。

一般情况下，创建一个类后并不能直接对类中的属性和方法进行引用，必须对类进行实例化，即创建一个对象。PHP 中用 new 关键字创建对象。

实例 511	图片的分栏显示 光盘位置：光盘\MR\11\511	高级 趣味指数：★★★★

实例说明

采用分栏的方式将多张图片同时显示在一个页面中，可以使页面更加规整。运行本实例，如图 11.7 所示。

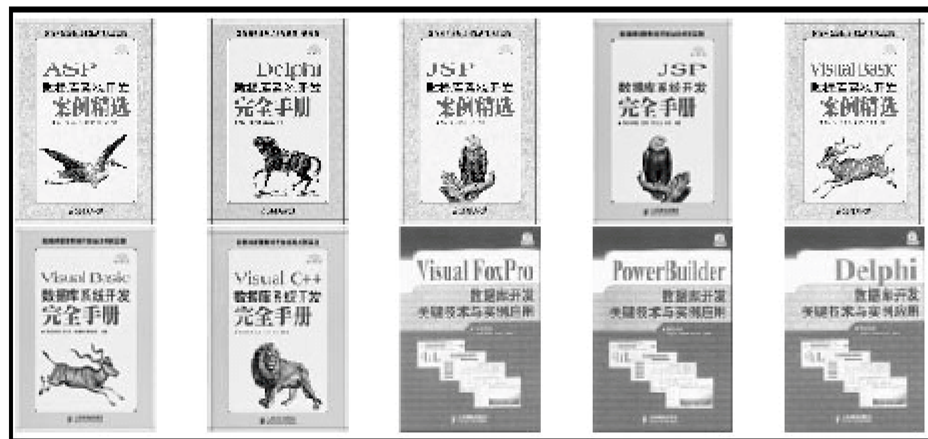


图 11.7　图片的分栏显示

关键技术

本实例实现图片的分栏显示主要依靠标记变量$i，如果该变量的值能被 5 整除则显示下一行（当然这是一栏显示 5 张图片的情况，如果一栏显示 4 张图片，可以通过判断该值能否被 4 整除，依次类推）。注意，显示一张图片后应使$i 的数值加 1，实现该过程的代码如下：

```
<?php
  $i=1;
	while(!$rs->eof){
		if ($i % 5==0){
	?>
		<td height="25">
		  <div align="left"><img src="<?php echo iconv('gbk','utf-8',$rs->fields(address)->value);?>" border="0"> </div>
	  </td>
	  </tr>
	  <?php
		 }else{
	?>
  <td height="25"><div align="left"><img src="<?php echo iconv('gbk','utf-8',$rs->fields(address)->value);?>" border="0"> </div>
  </td>
	<?php
		}
	  $rs->movenext;
	  $i++;
	}
 ?>
```

设计过程

（1）　建立与 Access 数据库的连接，代码如下：

```
<?php
    $conn=new com("adodb.connection");
    $connstr="provider=microsoft.jet.oledb.4.0;data
    source=".realpath("../data/db_database11.mdb ");
    $conn->open($connstr);
?>
```

（2） 查询数据库中要显示图片的地址并获得记录集，代码如下：

```
$sql="select * from tb_image ";
$rs=new com("adodb.recordset");
$rs->open($sql,$conn,1,3);
```

（3）实现图片的分栏显示，该部分代码已经在关键技术中给出，这里不再赘述。

秘笈心法

心法领悟 511：浅谈分栏显示。

所谓的分栏显示，就是取得标记变量的值，进行一些规定方法的运算，当运算结果符合条件时，便对标记变量的对应内容进行分栏。其实分栏操作只是编程世界中的一个小技巧，是程序员经验不断积累的结果。

实例 512 插入图书信息

光盘位置：光盘\MR\11\512

高级

趣味指数：★★★★

实例说明

为了让读者从多个方向了解和学习 Access 数据库，本实例将利用 ADODB 技术，对 Access 数据库进行操作，使读者在操作 Access 数据库方面，又多掌握了一种方法。运行本实例，如图 11.8 所示。

插入图书信息的实现

显示信息

输入字段信息每一项用空格分割 确定

图书ID	图书名称	作者	出版社	出版时间
3	PHP实战宝典	明日科技有限公司	人民邮电出版社	2010-08-13 00:00:00
4	PHP宝典	明日科技	北邮	2010-07-28 00:00:00
1	PHP网络编程	明日科技有限公司	人民邮电出版社	2010-08-12 00:00:00
2	PHP范例宝典	明日科技有限公司	人民邮电出版社	2010-08-12 00:00:00

图 11.8 插入图书信息

关键技术

本实例的关键点是灵活运用 ADODB 的相关函数，包括：

ADONewConnection()：连接数据库系统。

PConnect($host,[$user], [$password], [$database])：持久化连接数据库。

Execute($sql[,$inputarr=false])：执行 SQL 语句，并返回一个结果集（ADORecordSet 对象），如果失败则返回 FALSE。

本实例利用 ADODB 实现连接 Access 数据库的代码如下：

```
require_once("../adodb5/adodb.inc.php");                    //引入文件
$conn = ADONewConnection('access');                         //连接 Access 数据库
$conn -> Connect("Driver={Microsoft Access Driver (*.mdb)}; Dbq= F:\\AppServ\\www\\MR\\11\\data\\db_database11.mdb ");
```

有关 ADODB 技术的详细讲解，读者可以参考本书第 13 章的内容。

设计过程

（1）通过 ADODB 连接数据库，代码如关键技术中所示。

（2）编写“显示信息”按钮，单击此按钮，将输出 Access 数据库中存储的所有数据，代码如下：

```
<?php
if($_GET[id] == "1"){                                                    //单击按钮
        $rst = $conn -> execute("select * from tb_demo") or die ("出错");  //返回结果集
?>
<table bgcolor="#80E7EC" width="580px"><tr><td align="center">图书 ID</td><td align="center">图书名称</td><td align="center">作者</td><td
align="center">出版社</td><td align="center">出版时间</td></tr>
<?php
        while(!$rst -> EOF){                                             //如果指针没有指向最后一条数据
?>
<tr>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[0]);?></td>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[1]);?></td>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[2]);?></td>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[3]);?></td>
     <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst -> fields[4]);?></td>
</tr>
<?php
                $rst -> movenext();                                      //指针向下移动
        }
?>
```

（3）编写表单，通过字符串拆分函数，对文本框内容进行拆分，并执行插入操作，代码如下：

```
<?php
if($_POST[sub]){                                                         //单击按钮
        if($_POST[te] == "" || $_POST[te] == "输入字段信息每一项用空格分割"){    //判断文本框内容
                echo "<script>alert('请在文本框中输入内容');location.href='index.php?id=1';</script>";  //JavaScript1 提示
        }else{
                $array = explode(" ",$_POST[te]);                        //拆分字符串
                if(count($array) != 5){                                  //判断数组
                        echo "<script>alert('文本框输入内容有误,数据表共 5 个字段');location.href='index.php?id=1';</script>";  ///JavaScript 提示
                }else{
                        $a = iconv('utf-8','gbk',$array[0]);             //转码操作
                        $b = iconv('utf-8','gbk',$array[1]);
                        $c = iconv('utf-8','gbk',$array[2]);
                        $d = iconv('utf-8','gbk',$array[3]);
                        $e = iconv('utf-8','gbk',$array[4]);
                        //判断是否插入成功
                        if($conn -> execute("insert into tb_demo(`id`,`bname`,`author`,`pub`,`date`)values('".$a."','".$b."','".$c."','".$d."','".$e."')")){
                                echo "<script>alert('插入数据成功');location.href='index.php?id=1';</script>";  //JavaScript 提示
                        }else{
                                echo "<script>alert('SQL 语句有误，请重新输入');location.href='index.php?id=1';</script>";
                        }
                }
        }
}
?>
```

秘笈心法

心法领悟 512：Access 数据库的 SQL 语句。

Access 数据库与 MySQL 数据库的 SQL 语句，在更新、删除、查询方面几乎相同，但是在插入语句方面有所不同。在编辑插入的 insert 语句时，数据表字段必须由“`”包含。

实例 513	删除图书信息 光盘位置：光盘\MR\11\513	高级 趣味指数：★★★★

实例说明

Access 数据库虽然小巧，但是在功能上并不逊于其他数据库，进行简单的插入、删除操作是很迅速的。本实例通过 ADODB 连接 Access 数据库，实现删除图书信息的操作，运行结果如图 11.9 所示。



ADODB，全称为 Active Data Objects Data Base，是存取数据库所使用到的一组函数。虽然 PHP 是构建 Web 系统强有力的工具，但是由于 PHP 存取函数一直没有标准化，不同数据库间的函数名称、参数差异很大，在更换数据库时，会带来大量的代码修复工作。这时，就需要一组函数库来隐藏不同数据库函数间的差异，以让开发者简单去切换数据库，这就是 ADODB 类库。有关其详细的讲解请读者参考本书第 13 章。

11.3　Access 数据库的安全

数据库文件的安全非常重要，尤其是对于利用 Access 数据库建立的小型网站而言，用户总是担心数据库文件会不会被下载、用户名和密码是否会被他人盗走、系统是否会被人攻击等。下面通过具体实例来讲解网站的安全措施。

实例 514	操作带密码的 Access 数据库 光盘位置：光盘\MR\11\514	高级 趣味指数：★★★★

实例说明

为防止用户通过下载 Access 数据库文件来达到窃取机密信息的目的，可以为 Access 数据库设置密码，以保护数据，这样即使用户得到数据库，也会因为没有密码而无法查看数据库中的内容。运行本实例，对 Access 数据库设置密码的效果如图 11.10 所示，在网站中连接数据库的运行结果如图 11.11 所示。

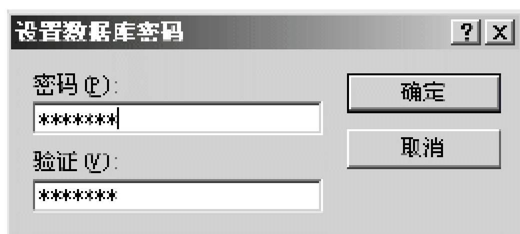


图 11.10　设置数据库密码

书名	出版社	出版时间	页数	价格
Delphi数据库开发关键技术与实例应用	人民邮电出版社	2004年5月	407 页	40 元
Delphi数据库开发实例解析	机械工业出版社	2004年2月	480 页	49 元
ASP数据库开发实例解析	机械工业出版社	2004年12月	321 页	36 元
Visual FoxPro数据库开发实例解析	机械工业出版社	2003年9月	458 页	48 元
Power Builder数据库开发实例解析	机械工业出版社	2003年9月	400 页	43 元
Visual Basic数据库开发实例解析	机械工业出版社	2003年8月	460 页	48 元
Visual Basic精彩编程200例	机械工业出版社	2003年1月	358 页	39 元

本站共有图书47种 每页显示10种 第5页/共5页　1 2 3 4　GO

图 11.11　操作带密码的 Access 数据库

关键技术

本实例主要通过 Access 菜单栏中的“工具”/“安全”/“设置数据库密码”命令来设置 Access 数据库的密码，同时也可以撤销数据库密码。

撤销数据库密码的过程为选择 Access 菜单栏中的“工具”/“安全”/“撤销数据库密码”命令，在弹出的“撤销数据库密码”对话框中输入设置时密码即可。

提示：在设置或者撤销 Access 数据库的密码时，必须将数据库以独占的方式打开。

设计过程

（1）启动 Access 数据库，单击工具栏中的按钮，弹出“打开”对话框，选择需要设置密码的数据库文件，例如 bookinfo.mdb，单击“打开”按钮右侧的下三角按钮，在弹出的菜单中选择“以独占方式打开”选项，则该数据库文件将以独占方式打开。

（2）选择主菜单上的“工具”/“安全”/“设置数据库密码”命令，将弹出如图 11.10 所示的对话框，在该对话框中输入密码及密码验证码，单击“确定”按钮即可完成密码的设置。

（3）采用 ADO 方式连接加密的 Access 数据库。注意，连接加密的 Access 数据库时，只需在数据库连接驱动中通过关键字 pwd 指定密码字符串即可，代码如下：

```
<?php
//利用 PHP 预定义类 com 声明一个数据库连接对象，并利用 ADO 连接数据库
$conn = new com("adodb.connection");
//设置数据库连接驱动
$connstr="driver={microsoft access driver (*.mdb)};pwd=mrsoft; dbq=". realpath("data/db_bookinfo.mdb");
$conn->open($connstr);                              //调用 com 类的 open()方法来执行上述连接驱动
?>
```

（4）完成图书信息分页显示，其关键代码如下：

```
<?php
    include("conn/conn.php");                       //包含数据库连接文件 conn.php
    $sql="select * from tb_bookinfo order by pdate desc";    //查询 tb_bookinfo 表中的所有图书信息
    $rs=new com("adodb.recordset");                 //用 com 类声明一个记录集对象
    $rs->open($sql,$conn,1,3);                      //利用 com 类的 open()方法执行查询
    $rs->pagesize=10;                               //设置每页最多显示 10 条记录
    if((trim(intval($_GET[page]))=="")||(intval($_GET[page])>$rs->pagecount)||(intval($_GET[page])<=0)){
          $page=1;
       }else{                                       //否则使$page 的值为所接收的查询字符串的数值
          $page=intval($_GET[page]);
       }
     if($rs->eof || $rs->bof) {                     //判断表中的内容是否为空，如果为空则给出提示
?>
<tr><td height="20" colspan="5" bgcolor="#FFFFFF"><div align="center">本站暂无图书信息！</div></td></tr>
<?php
       }else{                          //不为空则分页显示数据
        $rs->absolutepage=$page;       //设置当前显示页为所接收的查询字符串的值
        $mypagesize=$rs->pagesize;     //定义变量$mypagesize 用于控制当前页的循环终止
        while(!$rs->eof && $mypagesize>0) {  //记录集对象没有到整个表尾并且当前页所要显示的记录没有完全显示完，则循环显示数据信息
?>
      <tr>
          <td height="20" bgcolor="#FFFFFF"><div align="left"> <?php $fields=$rs->fields(bookname);echo $fields->value;?></div></td>
          <td height="20" bgcolor="#FFFFFF"><div align="center"><?php $fields=$rs->fields(tpi);echo $fields->value;?></div></td>
          <td height="20" bgcolor="#FFFFFF"><div align="center"><?php $fields=$rs->fields(pdate);echo $fields->value;?></div></td>
          <td height="20" align="center" bgcolor="#FFFFFF"><?php $fields=$rs->fields(bookpage);echo $fields->value;?> 页</td>
          <td height="20" align="center" bgcolor="#FFFFFF"><?php $fields=$rs->fields(price);echo $fields->value;?> 元</td>
       </tr>
   <?php
       $mypagesize--;                  //每显示一条记录，应使$mypagesize 的值减 1
       $rs->movenext;                  //使记录集指针移到下一条记录的位置
      }
    }
   ?>
</tr>
<tr>
     <td width="283" height="25"><div align="left">本站共有图书<?php echo $rs->recordcount;?>种 每页显示<?php echo $rs->pagesize;?>种 第<?php echo $page;?>页/共<?php echo $rs->pagecount;?>页</div></td>
     <td width="263"><div align="right">
<?php
    if($page>=2) {                     //如果当前显示页的页码大于 2，则显示首页和返回前一页链接
?>
   <a href="index.php?page=1" title="首页"><font face="webdings"> 9 </font></a>
   <a href="index.php?page=<?php echo $page-1;?>" title="前一页"><font face="webdings"> 7 </font></a>
<?php
     }
    if($rs->pagecount<=4) {            //如果总页数小于等于 4，则显示到所有页的链接
        for($i=1;$i<=$rs->pagecount;$i++){
?>
          <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
<?php
          }
   }else {                             //如果总页数大于 4，则显示到前 4 页的链接
```

```
        for($i=1;$i<=4;$i++) {
?>
            <a href="index.php?page=<?php echo $i;?>"><?php echo $i;?></a>
<?php
        }
?>
        <a href="index.php?page=<?php
  if($rs->pagecount>=$page+1)                    //如果到下页链接的页码大于总页数则显示第一页
     echo $page+1;
  else
     echo 1;
  ?>" title="后一页"><font face="webdings"> 8 </font></a>
 <a href="index.php?page=<?php echo $rs->pagecount;?>" title="尾页"><font face="webdings"> : </font></a>
   <?php
        }
   ?>
 </div>
 </td>
<form name="form1" method="get" action="index.php">
        <input type="text" name="page" size="2" class="inputcss">
        <input name="submit" type="submit" class="buttoncss" value="GO">
</form>
```

秘笈心法

心法领悟 514：支持跳转的分页显示功能。

在本实例中应用面向过程的技术编写分页方法，实现在不同页面之间的跳转、上一页和下一页之间的跳转以及跳转到指定页面的功能。

实例 515	防止 Access 数据库被下载 光盘位置：光盘\MR\11\515	高级 趣味指数：★★★★

实例说明

在开发中小型 Web 应用程序时，通常采用 Access 数据库，因为应用 Access 数据库比较方便。在站点上传时，通常将数据库文件和其他文件一同上传到网站服务器中，但是这样做是极不安全的，一旦被用户猜测到数据库的相对路径，用户在客户端便可以下载该数据库文件。例如，用户猜测到某网站的数据库保存在站点目录中的 db_news.mdb 文件中，在浏览器的地址栏中输入 http://127.0.0.1/mr/11/010/data/db_news.php 后，即可弹出如图 11.12 所示的文件下载对话框。

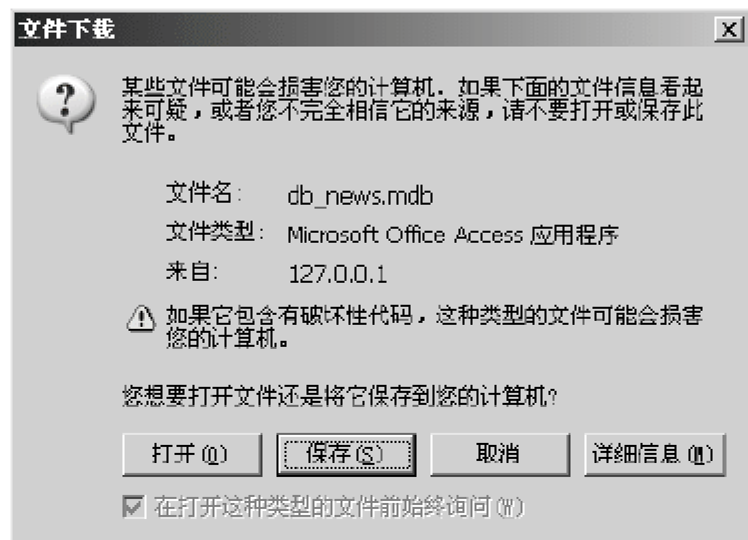


图 11.12　在客户端下载数据库文件

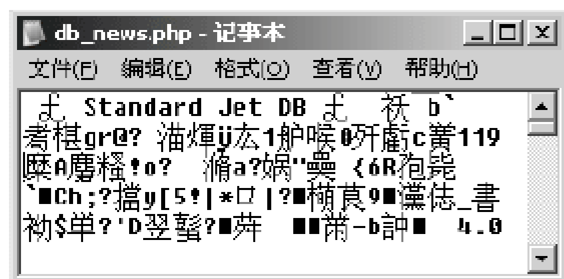


图 11.13　在记事本中打开的数据库文件

关键技术

解决以上问题的方法是将数据库文件的扩展名由.mdb 改为.php，即在连接数据库时，将数据库文件指定为.php 文件即可。此时再打开该文件（db_news.php）时，将会出现如图 11.13 所示的效果。在默认情况下，文

件夹内存放的文件的扩展名是不显示的，此时无法修改文件的扩展名。

设计过程

（1）修改实例中所用数据库的扩展名，将其命名为 db_news.php。

（2）新建 conn 文件夹，编辑 conn.php 文件，定义 Access 数据库的连接方法，其代码如下：

```
<?php
$conn = new com("adodb.connection");
$connstr="driver={microsoft access driver (*.mdb)}; dbq=". realpath("data/db_news.php");
$conn->open($connstr);
?>
```

（3）新建 index.php 文件，包含数据库连接文件，应用 while 语句完成数据库中数据的循环输出，其关键代码如下：

```
<?php
    include("conn/conn.php");                              //包含数据库连接文件
    $sql="select * from tb_news order by pdate desc";      //定义查询语句
    $rs=new com("adodb.recordset");                        //实例化类
    $rs->open($sql,$conn,1,3);                             //执行查询操作
    if($rs->eof || $rs->bof){                              //判断查询结果是否为真
?>
<tr><td height="20" colspan="5" bgcolor="#FFFFFF"><div align="center">暂无新闻信息！</div></td></tr>
<?php
        }else{
          while(!$rs->eof){                                //循环输出查询结果
  ?>
<tr>
      <td height="20" align="center" bgcolor="#FFFFFF"><div align="center">
          <?php $fields=$rs->fields(type);echo $fields->value;?>
                 </div></td>
      <td height="20" align="left" bgcolor="#FFFFFF"><div align="left">
          <?php $fields=$rs->fields(newobj);echo $fields->value;?>
      </div></td>
</tr>
<?php
        $rs->movenext;
       }
    }
?>
```

秘笈心法

心法领悟 515：Access 数据库的安全。

首先为 Access 数据库设置密码，然后更改 Access 数据库文件的后缀名称，通过这两项设置，相信 Access 数据库的安全可以得到一定的保障。

11.4　Access 数据库的综合运用

本节以 Access 数据库为数据的存储方式，开发一个聊天室功能模块，通过这个功能模块体会 Access 数据库在完整项目中的运用。

实例 516	ADODB 连接 Access 数据库 光盘位置：光盘\MR\11\516	高级 趣味指数：★★★★

实例说明

在本实例中介绍通过 ADODB 类库连接 Access 数据库，这里连接的 Access 数据库是具有密码的，并且隐

藏了 Access 数据库文件的后缀。运行本模块中的 top.php 文件，就可以测试出数据库是否连接成功，如果连接成功，将输出如图 11.14 所示的页面。



图 11.14　成功连接 Access 数据库

关键技术

这里应用 ADODB 类库连接 Access 数据库，并且 Access 数据库设置了密码，同时对数据库文件名称的后缀进行了修改。所以在封装 Access 数据库的连接类时，必须注意以下内容：

☑　要指定正确的 ADODB 类库的调用路径。

☑　由于使用 ADODB 类库操作 Access 数据库，数据库类型的判断应该使用 ado_access。

☑　定义连接语句时，不要忘记 Driver 与(*.mdb)之间的空格。

☑　Access 数据库的访问密码为 mrsoft。

☑　Access 数据库文件的后缀是.php。

设计过程

（1）定位 ADODB 类库的位置，将其存储于章目录 11 下，命名为 adodb5。

（2）定位 Access 数据库文件的位置，将其存储于 11\011\data 文件夹下，命名为 db_messages.php。

（3）在 11\011\conn 文件夹下新建 ConnDB.php 文件，封装 ADODB 连接 Access 数据库类，其代码如下：

```
<?php
class ConnDB{
	var $dbType;								//数据库类型标识
	var $host;								//数据库服务器地址
	var $userName;							//数据库服务器用户名
	var $pwd;								//数据库用户密码
	var $dbName;							//要连接的数据库名
	var $isDebug;							//是否显示调试信息
	var $connID;							//数据库连接 ID
	function ConnDB ($dbType = 'mysql', $host, $userName, $pwd, $dbName, $isDebug = false){
		$this->dbType = $dbType;					//为数据库类型赋初值
		$this->host = $host;					//为服务器地址赋初值
		$this->userName = $userName;				//为用户名赋初值
		$this->pwd = $pwd;					//为密码赋初值
		$this->dbName = $dbName;				//为数据库名赋初值
		$this->isDebug = $isDebug;				//为调试信息赋初值
	}
	function getConnID (){
		require_once '../adodb5/adodb.inc.php';			//导入 ADODB 类库
		$this->connID = NewADOConnection($this->dbType);	//生成数据库连接对象
		if ($this->dbType == 'mysql' || $this->dbType == 'mssql') { //如果为 MySQL 数据库或者 SQL Server 数据库
			$this->connID->Connect($this->host, $this->userName, $this->pwd, $this->dbName);	//通过 Connect()方法建立与数据库连接
			if ($this->dbType == 'mysql') {
				$this->connID->Execute('set names gb2312');	//设置数据库编码
			}
		} elseif ($this->dbType == 'ado_access') {		//如果为 Access 数据库
			//注意在连接 Access 数据库时，编写连接语句一定要注意空格的使用，必须严格执行，否则很可能导致连接失败
			$this->connID->Connect('Driver={Microsoft Access Driver (*.mdb)};Uid='.$this->userName.'; Pwd='.$this->pwd .';
Dbq='.realpath($this->dbName).';');
		} else {
			return false;
		}
		$this->connID->debug = $this->isDebug;			//是否调试
		return $this->connID;					//返回数据库连接对象
	}
	function closeConnID (){
		@$this->connID->Disconnect();				//关闭与数据库的连接
	}
}
```

（4）如果要通过封装的 ConnDB 类连接数据库，必须对类进行实例化，同时向类中传递参数，最终返回连接标识，该操作被存储于 config.inc.php 文件中。在该文件中完成数据库连接类的实例化，返回连接的标识，同



关键技术

用户登录操作的完成应用 AdminDB 类中的 executeSQL()方法，在该方法中可以执行查询、添加、更新和删除的操作，AdminDB.php 文件的代码如下：

```
<?php
class AdminDB{
    function executeSQL ($sql, $connID){
        $sqlType = strtolower(substr(trim($sql), 0, 6));                        //提取 SQL 语句的类型
        $rs = $connID->Execute($sql);                                            //执行 SQL 语句
        if ($sqlType == 'select') {                                              //如果是 select 查询
            $arrayData = $rs->GetRows();                                         //返回查询记录集
            if (count($arrayData) == 0 || $rs == false) {                        //如果没查询到或发生错误
                return false;                                                    //返回 false
            } else {                                                             //否则
                return $arrayData;                                               //返回记录集
            }
        } elseif ($sqlType == 'insert' || $sqlType == 'update' || $sqlType == 'delete') {  //如果执行插入、更新或删除语句
            return $rs;                                                          //返回语句执行状态，即成功返回 true，失败返回 false
        } else {
            return false;                                                        //如果不是上述查询，则返回 false
        }
    }
}
```

在实例 516 中已经介绍过，该类的实例化操作同样存储于 config.inc.php 文件中，返回对象为$adminDB。

设计过程

（1）创建 index.php 文件，设置聊天室的登录页面，通过 form 表单将用户登录信息提交到 saveuser.php 文件中进行处理，同时创建验证用户名是否被占用的按钮，调用 JavaScript 脚本中的 open_chknc()方法验证用户名是否可用，其关键代码如下：

```
<?php
    include_once("top.php");                                                     //包含头文件
?>
<form name="form1" method="post" action="saveuser.php" onsubmit="return chkinput(this)">
    <input type="text" name="usernc" size="15" class="inputcss" maxlength="50">
    <input type="button" value="是否已用" onClick="open_chknc(form1.usernc.value)" class="buttoncss">
    <select name="userhead" onchange="javascript:change_head()">
<?php
for($i=0;$i<=11;$i++){
?>
        <option value="<?php echo("images/head/".$i.".gif");?>">头像<?php echo $i?></option>
<?php
   }
?>
    </select>
    <input type="submit" name="submit" class="buttoncss" value="进入">  
    <input type="reset" class="buttoncss" value="重写">
</form>
```

（2）在 js/check.js 文件的 open_chknc()方法中，调用 chkusernc.php 文件，验证提交的用户名是否被占用，其关键代码如下：

```
<?php
$nc=trim($_GET[nc]);
if($nc==""){
    echo "请输入用户昵称!";
}else{
    $info=$adminDB->executeSQL("select * from tb_puser where usernc='".$nc."'",$conn);
    if(!$info){
        echo "恭喜您,该昵称未被占用!";
    }else{
        echo "对不起,该昵称已被占用!";
    }
}
?>
```

（3）新建 saveuser.php 文件，验证登录用户名是否被占用，如果用户名可用，将用户登录信息存储到指定数据表中，并且跳转到聊天室主页面，否则提示用户名不可用，或者用户登录失败，其关键代码如下：

```
<?php
include_once("conn/config.inc.php");                                    //包含数据库连接、操作类的实例化文件
$usernc=trim($_POST[usernc]);                                            //获取提交的用户名
$sql=$adminDB->executeSQL("select * from tb_puser where usernc='".$usernc."'",$conn);    //验证用户名是否被占用
if($sql){
   echo "<script language='javascript'>alert('对不起，该昵称已被其他用户使用!');history.back();</script>";
   exit;
}else{
   $logintime=date("Y-m-j H:i:s");                                       //定义时间
   $userhead=$_POST[userhead];
   $online=1;
   $ip=getenv("REMOTE_ADDR");                                            //获取 IP
   $email=trim($_POST[email]);                                           //获取用户提交的 email
   $mark=0;
   $nowtime=time();                                                      //定义时间戳
   $arr=$adminDB->executeSQL("insert into tb_puser(usernc,logintime,ip,email,mark,userhead,online,nowtime) values ('$usernc','$logintime','$ip',
'$email','$mark','$userhead','$online','$nowtime')",$conn);       //执行用户信息的添加操作
   if($arr){
        $_SESSION["nc"]=$usernc;                                         //用户登录成功后将用户名赋给 SESSION 变量
        echo "<script>window.location.href='chat.php';</script>";        //跳转到聊天室主页
   }else{
        echo "添加失败！";
   }
}
?>
```

秘笈心法

心法领悟 517：使用 getenv()函数获取客户端的 IP 地址。

在本实例中，应用 getenv()函数获取客户端的 IP 地址，其作用与$_SERVER['REMOTE_ADDR']是相同的。

实例 518	聊天室 光盘位置：光盘\MR\11\518	高级 趣味指数：★★★★

实例说明

聊天室的主页面由头文件 top.php、尾文件 bottom.php、发布聊天信息 chat.php、聊天内容 listchat.php 和聊天参与者 listuser.php 共 5 个部分组成，其运行效果如图 11.16 所示。



图 11.16 聊天室主页面



```
                        } else {
                                echo $_SESSION ["snc"];
                        }
                }
        }
?>">   
        <input type="submit" name="submit" value="发送" class="buttoncss" />  
        <input name="reset" type="reset" class="buttoncss" value="重写">
</form>
```

（2）新建 saveword.php 文件，将聊天信息存储到指定的数据表中，其代码如下：

```
<?php
include_once ("conn/config.inc.php");                                   //包含配置文件
$fromuser = trim ( $_SESSION ["nc"] );                                  //获取登录用户名称
$sql = $adminDB->executeSQL ( "select * from tb_puser where usernc='" . $fromuser . "'", $conn );    //判断用户是否存在
if (! $sql) {
        echo "<script>alert('对不起,等待时间过长，请重新登录!');window.location.href='logout.php';</script>";
        exit ();
}
$touser = $_POST ['touser1'];
if ($touser == "所有人") {
        $touser = "all";
}
$content = $_POST ['bbs_content'];
$pubtime = date ( "Y-m-j H:i:s" );
$nowtime = time ();
$arr = $adminDB->executeSQL ( "insert into tb_message(touser,fromuser,content,pubtime,nowtime) values ('$touser','$fromuser', '$content', '$pubtime',
'$nowtime')", $conn );                          //执行信息的添加操作
if ($arr) {
        echo "<script>window.location.href='chat.php';</script>";
} else {
        echo "添加失败！ ";
}
?>
```

（3）新建 listuser.php 文件，输出聊天室当前的在线用户，并通过 JavaScript 脚本中的 selectuser()方法调用 selectuser.php 文件，完成聊天对象的选择，其关键代码如下：

```
<script language="javascript">
function selectuser(x){
      parent.form_bbs.touser.value=x;
       parent.form_bbs.touser1.value=x;
       window.location.href='selectuser.php?usernc='+x;
}
</script>
<table width="120" border="0" align="center" cellpadding="0" cellspacing="0">
  <?php
 $array = $adminDB->executeSQL ( "select * from tb_puser where online=1 ", $conn );
 if (! $array) {
         ?>
   <tr><td height="25" colspan="2"><div align="center">暂无人登录</div></td></tr>
  <?php
 } else {
         ?>
  <tr onClick="javascript:selectuser('所有人')" style="cursor: hand">
         <td height="25" colspan="3" bgcolor="#99CCFF"><div align="center" class="STYLE1">发给所有参与者</div></td>
 </tr>

   <?php
         for($i = 0; $i < count ( $array ); $i ++) { //通过 for 循环输出数据库中的数据
                 if ($array [$i] [usernc] != $_SESSION ["nc"]) {
                         ?>
  <tr onClick="javascript:selectuser('<?php echo $array [$i] [1]; ?>')" style="cursor: hand">
         <td width="25" height="25"><div align="center"><img src="<?php echo $array [$i] [6]; ?>" width="20" height="20" onMouseOver=
"toolTip('<?php echo $array [$i] [1]; ?>', '<?php echo $array [$i] [3]; ?>', '<?php echo $array [$i] [4]; ?>', '<?php echo $array [$i] [2]; ?>');"
onmouseout= toolTip(); border=0></div></td>
         <td width="77" height="25"><div align="center"><?php echo $array [$i] [1]; ?></div>
         </td><?php
                 }
   }
```

```
 }
 ?>
</table>
```

（4）新建 selectuser.php 文件，通过 SESSION 定位聊天对象，代码如下：

```
<?php
session_start();                                    //初始化 SESSION 变量
unset ( $_SESSION['snc'] ) ;                        //删除 SESSION 变量中的值
$_SESSION["snc"]=$_GET[usernc];                     //根据超链接传递的变量值，重新为 SESSION 变量赋值
echo "<script>window.location.href='listuser.php';</script>";
?>
```

（5）新建 listchat.php 文件，完成聊天信息的循环输出，其关键代码如下：

```
<table width="500" border="0" align="center" cellpadding="0"cellspacing="0">
<?php
include_once ("function.php");
$sql = $adminDB->executeSQL ( "select * from tb_message ", $conn );
if ($sql) {
        for($i = 0; $i < count ( $sql ); $i ++) { //通过 for 循环输出数据库中的数据
                if ($sql [$i] [1] == "all") {
                ?>
   <tr>
        <td height="20"><img src="images/topic.gif" width="10" height="9"></td>
        <td><div align="left"><font color="#3333FF">[ <strong><?php echo $sql [$i] [4]; ?></strong> ] 对
                [  <strong>所有人</strong> ]说：   <?php echo getday ( $sql [$i] [3] ); ?> )</font></div>
        </td>
    </tr>
  <tr>
        <td height="25" colspan="2"><div align="left">    <?php echo unhtml ( $sql [$i] [2] ); ?></div></td>
 </tr>
  <?php
                } else {
                        if ($_SESSION ["nc"] == $sql [$i] [1] || $_SESSION ["nc"] == $sql [$i] [4]) {
                        ?>
 <tr>
        <td width="36" height="20"><img src="images/topic.gif" width="10" height="9"></td>
        <td><div align="left"><font color="#3333FF">[ <strong><?php echo $sql [$i] [4]; ?></strong> ] 对
        [  <strong><?php echo $sql [$i] [1]; ?></strong> ]  说：  ( <?php echo getday ( $sql [$i] [3] ); ?> )</font></div>
        </td>
 </tr>
 <tr>
        <td height="25" colspan="2"><div align="left">    <?php echo unhtml ( $sql [$i] [2] ); ?></div>
        </td>
 </tr>
   <?php
                        }
                }
        }
}
?>
</table>
```

（6）新建 function.php 文件，编辑自定义函数 unhtml()对聊天室中输出的聊天信息进行格式化；编辑 getday()函数计算时间。

秘笈心法

心法领悟 518：私聊功能的实现。

PHP 中实现私聊的方法有多种，用 SESSION 实现私聊相对简单，本聊天室中私聊的关键代码如下：

```
<?php
   if($_SESSION["nc"]==$info[touser] ||$_SESSION["nc"]==$info[fromuser] ){
?>
   …… //显示私聊内容
<?php
    }
?>
```

上述简短的代码说明了实现私聊的原理，$_SESSION["nc"]保存的是当前登录用户的昵称，$info[touser]表



```
$usernc=$_SESSION["nc"];
if($usernc!="" || $usernc=='tsoft' ){
$sql=$adminDB->executeSQL("delete from tb_message ",$conn);
 if($sql){
	echo "<script>alert('聊天记录删除成功！ ');window.location.href='chat.php';</script>";
 }
}else{
 echo "<script>alert('不允许非法操作');window.location.href='admin_login.php';</script>";
}
?>
```

（4）新建 logout.php 文件，执行用户或者管理员的退出功能，其代码如下：

```
<?php
include_once("conn/config.inc.php");
$usernc=$_SESSION["nc"];
if($usernc=="tsoft"){
	unset ( $_SESSION['snc'] ) ;
	unset ( $_SESSION['nc'] ) ;
	echo "<script>alert('退出聊天室');window.location.href='index.php';</script>";
}else{
	$sql=$adminDB->executeSQL("delete from tb_puser where usernc='".$usernc."'",$conn);
	if($sql){
		unset ( $_SESSION['snc'] ) ;
		unset ( $_SESSION['nc'] ) ;
		echo "<script>alert('退出聊天室');window.location.href='index.php';</script>";
	}
}
?>
```

秘笈心法

心法领悟 519：定时刷新聊天室的聊天信息和在线列表。

定时刷新聊天室的聊天信息和在线列表应用的是 meta 标签的 http-equiv 属性和 content 属性。通过 http-equiv 设置 refresh 执行刷新，通过 content 设置刷新的时间间隔，其代码如下：

```
<meta http-equiv="refresh" content="5" />
```

第12章

PHP 操作 Oracle 数据库

- 连接 Oracle 数据库
- 操作 Oracle 数据库
- Oracle 数据库应用
- Oracle 数据的导入和导出

12.1　连接 Oracle 数据库

本节将介绍在 PHP 中如何实现与 Oracle 数据库的连接，这里分别讲解通过 Oracle 函数库和 ADODB 连接、操作 Oracle 数据库。

实例 520　oci_connect()函数连接 Oracle 数据库

光盘位置：光盘\MR\12\520

高级

趣味指数：★★★★☆

实例说明

PHP 连接 Oracle 数据库的方法很多，这里介绍通过 PHP 中自带的 Oracle 函数库来完成对 Oracle 数据库的操作。在本实例中，应用 oci_connect()函数连接 Oracle 数据库，然后应用 oci_parse()函数执行查询语句，最终通过 oci_fetch_array()函数和 while 语句完成数据库中数据的循环输出，其运行结果如图 12.1 所示。

图 12.1　oci_connect()函数连接 Oracle 数据库

关键技术

在 PHP 的 Oracle 函数库中提供有完成对 Oracle 数据库操作的函数，这里只对本实例中应用的函数进行详细介绍。

（1）oci_connect()函数，建立一个到 Oracle 服务器的连接，返回一个连接标识符，其语法如下：

```
resource oci_connect( string username, string password [, string db [, string charset [, int session_mode]]] )
```

oci_connect()函数的参数说明如表 12.1 所示。

表 12.1　oci_connect()函数的参数说明

参　　数	说　　明
username	必选参数，指定数据库中的用户名
password	必选参数，指定数据库中用户的密码
db	可选参数，包含要连接的本地 Oracle 的实例或者在 tnsnames.ora 中条目的名字。如果没有指定该参数，PHP 使用环境变量 ORACLE_SID（Oracle 实例）或 TWO_TASK（tnsnames.ora）来确定连接哪一个数据库
charset	使用 Oracle 服务器 9.2 或更高版本时，可以指定 charset 参数，被用于新的连接。如果使用低于 9.2 版本的 Oracle 服务器，本参数将被忽略并以 NLS_LANG 环境变量替代
session_mode	该参数自版本 1.1 起可用并接受如下值：OCI_DEFAULT、OCI_SYSOPER 和 OCI_SYSDBA。如果指定 OCI_SYSOPER 或 OCI_SYSDBA 其中之一，oci_connect()函数将尝试使用外部认证信息建立特权连接。特权连接默认被禁止，如果要启用，需要将 oci8.privileged_connect 设置为 On





图 12.2　ADODB 操作 Oracle 数据库

说明： 因为 ADODB 属于第三方组件，所以在本书的光盘中没有提供 adodb 类库文件，需要读者自行下载并复制到本章的 12 文件夹下，并且命名为 adodb5，然后才可以运行本章的实例。

关键技术

使用 ADODB 连接数据库系统应用 ADONewConnection()函数；如果选择持久化连接数据库则使用 Pconnect()函数，如果选择非持久化连接则使用 Connect()函数。

（1）ADONewConnection()函数，用于连接数据库系统，其语法如下：

```
ADONewConnection($databaseType)
```

参数$databaseType 表示要连接的数据库系统的名称。例如：mysql、oci8 等。

（2）PConnect()函数，实现与数据库持久化连接。其语法如下：

```
PConnect($host,[$user],[$password],[$database])
```

参数说明：

$host：数据库系统的服务器所在地址。如果是本机操作，参数值为 localhost。

$user：数据库用户名。

$password：数据库密码。

$database：使用到的数据库。

（3）Connect()函数，实现与数据库非持久化连接，其语法如下：

```
Connect($host,[$user],[$password],[$database])
```

参数说明：

$host：数据库系统的服务器所在地址。如果是本机操作，参数值为 localhost。

$user：数据库用户名。

$password：数据库密码。

$database：使用到的数据库。

技巧： 持久化连接和非持久化连接的区别：持久化连接不用每次都创建新连接，可以增加程序的执行速度，但是有些数据库不支持此函数，那么就需要使用 Connect()函数来替代 PConnect()函数。

（4）应用 execute()函数执行 SQL 语句，应用 EOF 变量、while 语句和 MoveNext()函数完成数据的循环输出。有关 ADODB 类库中函数的详细讲解请参考第 13 章。

设计过程

（1）将下载的 ADODB 类库命名为 adodb5，存储于 12 文件夹下，并且在 12 文件夹下创建文件夹 521 作

为实例的根目录。

（2）在实例根目录 521 下创建 conn 文件夹，编写 conn.php 文件，应用 ADONewConnection()函数和 Connect()函数连接 Oracle 数据库，其代码如下：

```
<?php
include('../adodb5/adodb.inc.php');                              //载入 ADODB
$conn = ADONewConnection('oci8');                                //建立一个 oracle 连接
$conn->Connect("oracle","system","mrsoft");                     //连接 Oracle 数据库，用户名为 system，口令为 mrsoft
?>
```

（3）创建 index.php 文件，通过 include 语句包含数据库连接文件，完成数据库中数据的循环输出，并且创建 form 表单，将员工信息提交到 index_ok.php 文件中，完成员工信息的添加操作，其关键代码如下：

```
<?php
if($_GET['conn_id']=="1"){
 include_once 'conn/conn.php';                                   //载入数据库连接文件
 $sqlstr = 'select * from tb_user ';                             //sql 查询语句
 $rst = $conn->execute ( $sqlstr ) or die ( 'error:   ' . $conn->errorMsg () );   //执行查询语句
 while ( ! $rst->EOF ) {                                         //wihle 语句循环输出结果
?>
<tr>
 <td align="center" bgcolor="#FFFFFF"><?php echo $rst->fields ['0'];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst->fields ['1']);?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst->fields ['2']);?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst->fields ['3']);?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo iconv("gb2312","utf-8",$rst->fields ['4']);?></td>
</tr>
<?php
         $rst->movenext ();                                      //指针下移
 }
 $rst->close ();                                                 //关闭连接
 $conn->close ();
}
?>
```

（4）创建 index_ok.php 文件，获取表单中提交的数据，将员工信息添加到 Oracle 数据库中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" );             //设置文件编码格式
include("conn/conn.php");
if($_POST['name']!=""){
$sql="insert into tb_user(id,name,position,age,dates)
values(idsid.nextval,'".iconv("utf-8","gb2312",$_POST['name'])."','".iconv("utf-8","gb2312",$_POST['position'])."','".iconv("utf-8","gb2312",$_POST['a
ge'])."','".iconv("utf-8","gb2312",$_POST['dates'])."')";
      $rst = $conn->execute ( $sql ) or die ( 'error:   ' . $conn->errorMsg () );      //执行添加语句
      if($rst){
           echo "<script>alert('信息添加成功！');window.location.href='index.php?conn_id=1';</script>";
      }else{
           echo "<script>alert('信息添加失败！');windwo.location.href='index.php'</script>";
      }
}else{
      echo "<script>alert('请输入员工信息！');windwo.location.href='index.php'</script>";
}
?>
```

秘笈心法

心法领悟 521：ADODB 连接 Oracle 数据库的关键。

（1）必须注意，在运用 ADONewConnection()函数时，指定的参数是 oci8。

（2）在 Connect()函数中，包括 3 个参数：第 1 个参数是数据库名称；第 2 个参数是用户名；第 3 个参数是口令。

（3）在输出数据库中数据时，需要应用 iconv()函数对输出的数据进行编码格式转换，由原来的 GB2312 编码转换为 utf-8 编码。



设计过程

这里在实例 521 的基础上，增加删除数据库中数据的功能，其具体的方法如下：

（1）在 index.php 文件中，为输出的每条数据添加一个删除超链接，链接到 delete_ok.php 文件，将数据的 ID 作为参数传递到 delete_ok.php 文件中，其关键代码如下：

```
<a href="delete_ok.php?conn_id=<?php echo $rst->fields ['0']; ?>">删除</a>
```

（2）创建 delete_ok.php 文件，以超链接传递的参数值为条件，执行删除操作，删除数据表指定的记录，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" );                    //设置文件编码格式
include("conn/conn.php");                                                //包含数据库连接文件
$sql="delete from tb_user where id='".$_GET['conn_id']."'";              //定义删除语句
$rst = $conn->execute ( $sql ) or die ( 'error：  ' . $conn->errorMsg () );   //执行删除语句
if($rst){
        echo "<script>alert('信息删除成功！');window.location.href='index.php';</script>";
}else{
        echo "<script>alert('信息删除失败！');windwo.location.href='index.php'</script>";
}
?>
```

秘笈心法

心法领悟 522：ADODB 类库。

在本章中只是应用 ADODB 类库连接、操作 Oracle 数据库，其实 ADODB 连接和操作任何数据库的函数、方法都是相同的，有关 ADODB 类库的具体运用将在第 13 章中进行详细的讲解。

12.2　操作 Oracle 数据库

PHP 对 Oracle 数据库有很好的函数支持，用户可以通过这些函数像操作 MySQL 数据库一样操作 Oracle 数据库。

实例 523	员工信息的添加、更新和查询 光盘位置：光盘\MR\12\523	高级 趣味指数：★★★☆

实例说明

本实例应用 PHP 和 Oracle 数据库创建一个企业用户管理模块，实现用户信息的添加、查询和更新操作。运行结果如图 12.4 所示。

图 12.4　企业用户管理操作页面









```
</tr>
<?php
}
?>
<tr>
        <td height="35" colspan="5" align="right" bgcolor="#FFFFFF"><span class="STYLE1"><?php echo "总共有$nrows  记录!";?> </td>
</tr>
<?php
}else{
?>
<tr><td colspan="5" align="center" bgcolor="#FFFFFF">无数据</td>
</tr>
<?php
}
oci_close($conn);
?>
```

秘笈心法

心法领悟 525：oci_fetch_all()函数的应用总结。

应用 oci_fetch_all()函数获取数据库中总的记录时，在其返回的数组中不但包括数据表中的数据，而且还包括数据表字段的名称。因此，在应用 oci_fetch_all()函数循环输出数据表中的数据时，不必单独定义数据表字段的名称表格，可以直接从数组中获取，同时也不必再应用其他函数对数据表中总的记录数进行统计。

实例 526　动态获取数据表中的字段和数据

光盘位置：光盘\MR\12\526

高级

趣味指数：★★★★☆

实例说明

如果采用固定字段名称的方法输出数据表中的数据，那么一旦数据表中的字段发生变化，再通过原来的表格输出数据表中的数据时，其运行效果将如图 12.7 所示，页面输出内容是不完整的。



图 12.7　采用固定字段名称的方法输出数据表中的数据

如果采用动态读取数据表中字段和数据的方法就可以避免图 12.7 所示的问题，无论字段发生怎样的变化，在页面中输出的数据都是完整的，不必对页面的设计进行重新调整，其运行效果如图 12.8 所示。



图 12.8　统计数据库中数据的记录总数

关键技术

在本实例中，首先应用 oci_num_fields()函数获取数据表中数据的列数，其语法如下：

```
int oci_num_fields( resource statement )
```

该函数返回参数 statement 中列的数目，其中参数 statement 为 oci_parse()函数返回的标识符。

然后，应用 oci_field_name()函数动态获取数据表中的字段信息，其语法如下：

```
string oci_field_name( resource statement, int field )
```

该函数返回与字段数字索引（从 1 开始）相对应的字段名。

最后，应用 oci_fetch_row()函数完成数据库中数据的循环输出，以数字作为数组的索引，其语法如下：

```
array oci_fetch_row( resource statement )
```

该函数从和指定的结果标识关联的结果集中获取一行数据并作为数组返回，如果没有更多行则返回 FALSE。每个结果的列储存在一个数组的单元中，数字索引从 0 开始。

通过本实例中介绍的函数循环输出数据库中的数据，可以避免因数据表增加字段或者减少字段，而要重新设计页面的麻烦。

设计过程

（1）新建 conn 文件夹，编写 conn.php 文件，定义 Oracle 数据库的连接方法，设置数据库的编码格式为 utf-8。

（2）新建 index.php 文件，首先包含数据库连接文件，然后应用 oci_num_fields()和 oci_field_name()函数以及 for 语句完成数据表中字段名称的输出。最后，应用 oci_fetch_row()函数和 while 语句根据数据表中字段的列数完成数据的循环输出，其关键代码如下：

```
<?php
include("conn/conn.php");                                   //包含数据库连接文件
$stmt = oci_parse($conn, "select * from tb_user");          //定义查询语句
oci_execute($stmt);                                         //预置 Oracle 语句
$ncols = oci_num_fields($stmt);                             //获取数据表字段的列数
for ($i = 1; $i <= $ncols; $i++) {                          //执行 for 循环
 $field_name   = oci_field_name($stmt, $i);                 //获取字段的名称
      echo "<td bgcolor='#FFFFFF' align='center' >$field_name</td>";
}
oci_free_statement($stmt);                                  //释放关联于 Oracle 语句或游标的所有资源
?>
              </tr>
<?php
$query="select * from tb_user ";                            //定义 SQL 语句
$result=oci_parse($conn,$query);                            //预置 Oracle 语句
$r = oci_execute($result, OCI_DEFAULT);                     //执行查询语句
while($row=oci_fetch_row($result)){                         //while 循环输出查询结果数组中的数据
?>
              <tr>

<?php
for ($i = 0; $i < $ncols; $i++) {                           //根据字段的列数，输出数据
      echo "<td bgcolor='#FFFFFF' align='center' >$row[$i]</td>";
}

?>
              </tr>
<?php
}
oci_close($conn);                                           //关闭数据库
?>
```

秘笈心法

心法领悟 526：盘点本实例应用的函数。

oci_field_name()函数返回与字段数字索引（从 1 开始）相对应的字段名，即获取第一个字段的名称时，应

该使用如下格式：

```
$field_name   = oci_field_name($stmt, 1);                    //获取字段的名称
```

oci_fetch_row()函数返回数组，数组的数字索引从 0 开始，即获取 oci_fetch_row()函数返回数组中的第一个元素值的格式是：

```
$row=oci_fetch_row($result);
$row[0];                                                     //数组中的第一个元素
```



实例 527　获取数据表中字段的详细信息

光盘位置：光盘\MR\12\527

高级

趣味指数：★★★★☆

实例说明

前面讲解的都是获取 Oracle 数据库中数据的方法，在本实例中将介绍如何获取数据表中字段的详细信息，运行结果如图 12.9 所示。

TB_USER2表的字段信息					
字段名称	字段精度	字段范围	数据类型	字段大小	原Oracle 数据类型
ID	38	0	NUMBER	22	2
NAME	0	0	VARCHAR2	40	1
POSITION	0	0	VARCHAR2	40	1
AGE	0	0	VARCHAR2	40	1
DATES	0	0	VARCHAR2	40	1

图 12.9　获取数据表中字段的详细信息

关键技术

本实例是多个函数的综合应用，其中包括：

☑ oci_num_fields()函数，获取结果列的数目。

☑ oci_field_name()函数，获取字段的名称。

☑ oci_field_precision()函数，获取字段的精度。

☑ oci_field_scale()函数，获取字段的范围。

☑ oci_field_type()函数，获取字段的数据类型。

☑ oci_field_size()函数，获取字段的大小。

☑ oci_field_type_raw()函数，获取字段的原始 Oracle 数据类型。

设计过程

本实例综合运用 Oracle 函数库中操作数据表字段的函数，获取数据表中字段的各种信息，并且以表格的形式输出获取到的数据，其关键代码如下：

```
<table width="580" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#339999">
  <tr><td colspan="6" align="center" bgcolor="#FFFFFF">TB_USER2 表的字段信息</td></tr>
  <tr>
    <td align="center" bgcolor="#FFFFFF">字段名称</td>
    <td align="center" bgcolor="#FFFFFF">字段精度</td>
    <td align="center" bgcolor="#FFFFFF">字段范围</td>
    <td align="center" bgcolor="#FFFFFF">数据类型</td>
    <td align="center" bgcolor="#FFFFFF">字段大小</td>
    <td align="center" bgcolor="#FFFFFF">原 Oracle  数据类型</td>
  </tr>
  <?php
    include("conn/conn.php");
    $stmt = oci_parse($conn, "select * from tb_user");               //定义查询语句
    oci_execute($stmt);
    $result = oci_num_fields($stmt);                                 //统计字段的数目
    for ($i = 1; $i <= $result; $i++) {
```

```
            $column_name    = oci_field_name($stmt, $i);              //获取字段名称
            $column_precision = oci_field_precision($stmt, $i);       //获取字段的精度
            $column_scale =oci_field_scale($stmt, $i);                //获取字段的范围
            $column_type    = oci_field_type($stmt, $i);              //字段的数据类型
            $column_size    = oci_field_size($stmt, $i);              //获取字段的大小
            $column_type_raw    = oci_field_type_raw($stmt, $i);      //获取字段的原始 Oracle 数据类型
        ?>
<tr>
      <td align="center" bgcolor="#FFFFFF"><?php echo $column_name ;?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $column_precision ;?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $column_scale ;?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $column_type ;?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $column_size ;?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $column_type_raw ;?></td>
   </tr>
<?php
      }
      oci_free_statement($stmt);                                      //释放资源
      oci_close($conn);                                               //关闭数据库
?>
</table>
```

秘笈心法

心法领悟 527：获取数据表中字段信息的函数。

获取数据表中的字段信息并不是一项常用的技术，它只会出现在一些特殊操作中，例如动态创建数据表、在网站的后台管理系统中查看数据表的结构等。其中只有 oci_num_fields ()和 oci_field_name()函数可能会经常使用，有关这些函数的详细讲解请参考 PHP 手册。

实例 528　修改 Oracle 数据库用户的密码

光盘位置：光盘\MR\12\528

高级　趣味指数：★★★★☆

实例说明

在本实例中，介绍一种可以远程修改 Oracle 数据库用户密码的方法，其运行效果如图 12.10 所示。



图 12.10　修改 Oracle 数据库用户的密码

关键技术

修改 Oracle 数据库用户的密码应用的是 oci_password_change()函数，其语法如下：

```
bool oci_password_change ( resource connection, string username, string old_password, string new_password )
```

oci_password_change()函数用于修改用户 username 的密码。如果成功返回 TRUE，失败则返回 FALSE。

oci_password_change()函数的参数说明如表 12.7 所示。

表 12.7　oci_password_change()函数的参数说明

参　　数	说　　明
connection	必选参数，数据库连接成功后返回的连接标识符
username	必选参数，数据库的用户名
old_password	必选参数，旧密码
new_password	必选参数，新密码

设计过程

（1）新建数据库连接文件 conn.php，连接用户名为 system，密码为 mrsoft 的 Oracle 数据库。

（2）新建 index.php 文件，创建 form 表单，将 Oracle 数据库的信息提交到 index_ok.php 文件中，在该文件中完成数据库用户密码的更新。

（3）创建 index_ok.php 文件，获取表单提交的 Oracle 数据库用户名的信息，应用 oci_password_change()函数更新 Oracle 数据库用户的密码，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" );                                    //设置文件编码格式
include("conn/conn.php");                                                               //包含数据库连接文件
if($_POST['user']!="" || $_POST['pass_old']!="" || $_POST['pass_new']==$_POST['pass_news']){
    $pass=oci_password_change($conn,$_POST['user'],$_POST['pass_old'],$_POST['pass_new']);   //执行更新操作
    if($pass==true){
        echo "<script>alert('密码更新成功！'); window.location.href='index.php';</script>";
    }else{
        echo "<script>alert('密码更新失败！'); window.location.href='index.php';</script>";
    }
    oci_close($conn);                                                                   //关闭数据库
}else{
 echo "<script>alert('添加信息不正确失败！'); window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 528：oci_password_change ()函数的第二种语法格式。

oci_password_change ()函数的第二种语法格式如下：

```
resource oci_password_change ( string dbname, string username, string old_password, string new_password )
```

该语法格式自版本 1.1（OCI8 扩展库版本）起可用。

实例 529　应用事务处理技术确保操作的同步执行　高级

光盘位置：光盘\MR\12\529　趣味指数：★★★★☆

实例说明

PHP 的事务处理机制可以确保在操作过程中遇到意外（突然断电、电脑死机等）时数据不丢失。在本实例中，应用事务实现数据删除和添加同时操作，并且输出操作执行后的结果，如图 12.11 所示。

图 12.11　事务处理机制的应用

关键技术

在 Oracle 数据库中通过 oci_commit()函数提交未执行的事务处理操作。oci_commit 函数的语法如下：

```
bool oci_commit( resource connection )
```

oci_commit()函数将正在运行的事务中所有未执行的查询语句提交处理，如果成功返回 TRUE，否则返回 FALSE。

通过 oci_rollback 函数回滚未提交的事务，其语法如下：

```
bool oci_rollback( resource connection )
```

oci_rollback()函数回滚 Oracle 连接 connection 上所有未提交的语句，如果成功返回 TRUE，否则返回 FALSE。

设计过程

（1）新建 conn.php 文件，完成与 Oracle 数据库的连接，并且设置数据库编码格式为 utf-8。

（2）新建 index.php 文件。首先，执行查询操作，并且将查询结果作为 form 表单元素的初始值，将表单元素中的值提交到 index_ok.php 文件中进行删除处理。然后，再次执行查询操作，输出执行删除操作后，数据库中剩余的数据，其关键代码如下：

```
<?php
include("conn/conn.php");
$query="select * from tb_user ";
$result=oci_parse($conn,$query);
$r = oci_execute($result, OCI_DEFAULT);
while($row=oci_fetch_array($result,OCI_RETURN_NULLS)){
?>
<form id="form1" name="form1" method="post" action="index_ok.php">
  <tr>
    <td align="center" valign="middle"><span class="STYLE4">
      <input name="id" type="text" id="id" size="5" value="<?php echo $row[0];?>" />
    </span></td>
<td align="center" valign="middle"><span class="STYLE4">
  <input name="user" type="text" id="user" size="8" value="<?php echo $row[1];?>" />
</span></td>
<td align="center" valign="middle"><span class="STYLE4">
  <input name="position" type="text" id="position" size="8" value="<?php echo $row[2];?>" />
</span></td>
<td align="center" valign="middle"><span class="STYLE4">
  <input name="age" type="text" id="age" size="10" value="<?php echo $row[3];?>" />
</span></td>
    <td align="center" valign="middle">
      <input type="submit" name="Submit" value="删除" />     </td>
    </tr>
  </form>
<?php }oci_close($conn);?>
```

（3）创建 index_ok.php 文件，获取 form 表单提交的数据，首先根据提交的 ID 值执行数据的删除操作，然后定义添加语句将表单中提交的数据添加到指定的数据表中。最后，应用 oci_commit()函数通过事务提交数据，如果提交执行失败，则应用 oci_rollback()函数实现事务的回滚，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" );                //设置文件编码格式
include("conn/conn.php");                                          //包含连接数据库文件
if($_POST['Submit']==true){
        $query="delete from tb_user where id='".$_POST['id']."'";   //定义删除语句
        $result=oci_parse($conn,$query);                           //预置 Oracle 语句
        $r=oci_execute($result,OCI_DEFAULT);                       //执行删除操作
        $query="insert into tb_user (id,name,position,age)values('1000','".$_POST['user']."','".$_POST['position']."','".$_POST['age']."')";
        $result=oci_parse($conn,$query);
        oci_execute($result, OCI_DEFAULT);                         //设置参数为 OCI_DEFAULT，停止语句的自动提交
        $committed = oci_commit($conn);                            //应用事务提交数据
        if (!$committed) {                                         //如果事务执行失败
                $error = oci_error($conn);
                echo '事务执行失败: ' . $error['message'];
        }
        oci_rollback($conn);                                       //事务回滚
```

```
        if($r){
                echo "<script> alert('删除成功!'); window.location.href='index.php';</script>";
        }
        oci_close($conn);
}
?>
```

秘笈心法

心法领悟 529：应用事务处理机制的注意事项。

（1）在应用 oci_commit()函数之前，应该先取消 oci_execute()函数的自动提交功能，否则事务将不会被执行。

（2）当关闭连接或脚本结束时事务会自动回滚。需要明确地调用 oci_commit()函数来提交事务，或调用 oci_rollback()函数来中止事务。

实例 530 通过 PHP 变量输出数据表中数据

光盘位置：光盘\MR\12\530

高级

趣味指数：★★★★☆

实例说明

在操作 Oracle 数据库时，循环输出数据库中的数据多数应用的是 Oracle 数据库函数。在本实例中，介绍一种特殊的输出数据的方法——将数据通过 PHP 变量进行输出，其运行结果如图 12.12 所示。

图 12.12　数据的循环输出

关键技术

本实例循环输出数据库中数据应用的是 oci_define_by_name()函数，将 PHP 变量值与 SQL 查询语句返回的列进行绑定，进而直接通过 PHP 变量输出数据库中的数据。

oci_define_by_name 函数，将 PHP 变量与 select 语句中返回的列进行绑定，如果定义的变量不存在于 select 语句中，并没有错误信息输出。该函数的语法如下：

```
bool oci_define_by_name ( resource statement, string column_name, mixed &variable [, int type] )
```

oci_define_by_name()函数的参数说明如表 12.8 所示。

表 12.8　oci_define_by_name()函数的参数说明

参　　数	说　　明
statement	必选参数，获取的语句标识符
column_name	必选参数，指定的字段名，必须大写
variable	必选参数，输出的变量

oci_fetch()函数，获取（对于 select 语句）结果中的下一行到内部结果缓冲区。如果成功则返回 TRUE，失败则返回 FALSE，其语法如下：

```
bool oci_fetch( resource statement )
```

在 PHP 5.0.0 之前的版本必须使用 ocifetch()替代本函数。ocifetch()函数名仍然可用，为向下兼容作为 oci_fetch()的别名，但不推荐使用。

设计过程

（1）新建 conn.php 文件，连接 Oracle 数据库，指定数据库编码格式为 utf-8，其代码如下：

```
<?php
//连接数据库名为 oracle，用户名是 system，密码是 mrsoft
//如果 oracle 数据库服务器与 PHP 程序处于同一机器中，在连接时数据库名称可以省略
$conn=oci_connect("system","mrsoft","oracle","utf8") or dir(oci_error());
?>
```

（2）新建 index.php 文件，通过 PHP 变量完成数据的循环输出。首先，包含数据库连接文件。然后，定义 SQL 语句，通过 oci_define_by_name()函数将 SQL 语句中查询出的字段名称与指定的变量进行绑定。最后，执行查询操作，通过 while 语句循环输出查询结果，其关键代码如下：

```
<?php
   include("conn/conn.php");                                      //包含数据库连接文件
     $query="select name, position, age from tb_user ";          //定义 SQL 语句
     $result=oci_parse($conn,$query);                            //预置 Oracle 语句
     oci_define_by_name($result, "NAME", $name);                 //将字段 name 与变量$name 绑定
     oci_define_by_name($result, "POSITION", $position);
     oci_define_by_name($result, "AGE", $age);
     oci_execute($result);                                       //执行查询操作
     while (oci_fetch($result)) {                                //循环输出查询结果
 ?>
  <tr>
    <td align="center" valign="middle"> </td>
      <td align="center" valign="middle"><span class="STYLE6"><?php echo $name;?></span></td>
<td align="center" valign="middle"><span class="STYLE6"><?php echo $position;?></span></td>
<td align="center" valign="middle"><span class="STYLE6"><?php echo $age;?></span></td>
  </tr>
<?php
}
    oci_free_statement($result);                                 //释放空间
    oci_close($conn);                                            //关闭数据库
?>
```

秘笈心法

心法领悟 530：应用 oci_define_by_name()函数的注意事项。

（1）Oracle 使用大写字母的列名，而用户在 select 中可以用小写字母；（2）如果需要定义一个抽象数据类型（LOB/ROWID/BFILE），必须先用 oci_new_descriptor()函数分配空间。

12.3　Oracle 数据库应用

前面只是讲解连接 Oracle 数据库的方法以及一些简单的操作，在本节中将 Oracle 数据库应用到实际的程序开发中。

实例 531	Oracle——管理员注册功能 光盘位置：光盘\MR\12\531	高级 趣味指数：★★★★

实例说明

本实例实现一个管理员的注册功能，应用 ADODB 连接、操作 Oracle 数据库，将注册信息添加到指定的

Oracle 数据表中，其运行结果如图 12.13 所示。



图 12.13　管理员注册

关键技术

有关 ADODB 连接、操作 Oracle 数据库的方法可以参考实例 521 和实例 522，这里不再赘述。

本实例在完成管理员注册信息的提交时，应用 GD2 函数生成一个数字验证码，防止数据的重复提交。其中验证码生成方法如 ValidatorCode.php 文件所示：

```
<?php
header('content-type:image/png');
srand((double) microtime() * 1000000);                       //定义随机数种子
$im = imagecreate(65, 25);                                   //创建画布
imagefill($im, 0, 0, imagecolorallocate($im, 200, 200, 200));  //定义画布背景
$validatorCode = $_GET['code'];                              //获取验证码的值
//在画布中输出验证码的值
imagestring($im, 8, 9, 2, substr($validatorCode, 0, 1), imagecolorallocate($im, 0, rand(0, 255), rand(0, 255)));
imagestring($im, 8, 22, 4, substr($validatorCode, 1, 1), imagecolorallocate($im, rand(0, 255), 0, rand(0, 255)));
imagestring($im, 8, 36, 6, substr($validatorCode, 2, 1), imagecolorallocate($im, rand(0, 255), rand(0, 255), 0));
imagestring($im, 8, 48, 8, substr($validatorCode, 3, 1), imagecolorallocate($im, 0, rand(0, 255), rand(0, 255)));
//添加干扰线
for ($i = 0; $i < 200; $i ++) {
    imagesetpixel($im, rand() % 70, rand() % 30, imagecolorallocate($im, rand(0, 255), rand(0, 255), rand(0, 255)));
}
imagepng($im);                                               //生成.png 图像
imagedestroy();                                              //销毁图像
```

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，载入 adodb.ini.php 文件。实现与 Oracle 数据库服务器中 Oracle 数据库的连接，用户名是 system，口令（密码）是 mrsoft，其代码如下：

```
<?php
include('../adodb5/adodb.inc.php');                          //载入 ADODB
$conn = ADONewConnection('oci8');                            //建立一个 oracle 连接
$conn->Connect("oracle","system","mrsoft");
?>
```

（2）创建 index.php 文件。首先，通过 script 标签载入 js 脚本文件。然后，创建 form 表单，将管理员注册信息提交到 index_ok.php 文件中，同时定义 onSubmit 事件调用 chkinput()方法对表单中元素值进行验证。最后，创建 JavaScript 脚本，定义方法完成验证码的输出和刷新操作，其关键代码如下：

```
<script language="javascript" src="js/check.js"></script>
<form id="form1" name="form1" method="post" onSubmit="return chkinput(this)" action="index_ok.php">
<table width="480" border="0" align="center" cellpadding="0" cellspacing="0">
<tr>
      <td width="87" align="right">管理员：</td>
        <td width="380" align="left"><input name="name" type="text" id="name" size="30" /></td>
    </tr>
```





图 12.14　管理员登录

关键技术

本实例同样应用 ADODB 连接、操作 Oracle 数据库，在判断管理员名称和密码时，应用 GetArray()函数返回查询结果集数组，应用 count()函数统计数组中元素个数，进而完成对登录管理员的判断。

GetArray([$number_of_rows])函数，返回从当前指针指向的记录开始，到（$number_of_rows – 1）行的全部记录的数组。

参数$number_of_rows 表示指定的记录行。如果没有给出，则一直到 EOF 才停止。

count()函数，返回数组中的单元数目，用来计算数组中值的个数，其语法如下：

```
int count ( mixed array [, int mode])
```

参数说明：

array：必选参数，指输入的数组。

mode：可选参数，COUNT_RECURSIVE（或 1）。如选中此参数，本函数将递归地对数组计数，对计算多维数组的所有单元尤其有用。此参数的默认值是 0。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，实现与 Oracle 数据库的连接。

（2）创建 index.php 文件，添加 form 表单，将管理员登录信息提交到 index_ok.php 文件中。

（3）创建 index_ok.php 文件，根据表单中提交的管理员名称和密码执行查询语句，如果返回值大于 0，则说明名称和密码正确，将跳转到 main.php 页面；否则将弹出提示信息，并跳转到 index.php 页面，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" ); //设置文件编码格式
include("conn/conn.php");
if($_POST['user']!="" && $_POST['pass']!=""){
$sql="select * from tb_admin where name='".iconv("utf-8","gb2312",$_POST['user'])."' and pass='".md5(iconv("utf-8","gb2312",$_POST['pass']))."'";
$rst = $conn->execute ( $sql ) or die ( 'error：  ' . $conn->errorMsg () );
if(count($rst->GetArray())>0){
        echo "<script>alert('管理员登录成功！ ');window.location.href='main.php';</script>";
}else{
        echo "<script>alert('管理员登录失败！ ');window.location.href='index.php'</script>";
}
}else{
    echo "<script>alert('请输入管理员信息！ ');window.location.href='index.php'</script>";
}
?>
```

（4）创建 main.php 文件，定义管理员登录成功后展示的页面。

秘笈心法

心法领悟 532：编码格式转换的必要性。

在创建的 Oracle 数据库中，数据的编码格式是 gb2312，而程序页面中的编码格式是 utf-8，所以在执行 select 查询语句时，在 where 条件中，必须应用 iconv()函数对表单中提交的用户名和密码进行编码转换，使其与数据库中存储的数据编码统一，这样才可以进行比较。否则由于数据的编码格式不同，无论输入的用户名和密码是否正确都不能成功登录。唯一一种对数据编码格式没有限制的情况就是数据中不包含中文字符串。

实例 533	Oracle——管理员数据管理 光盘位置：光盘\MR\12\533	高级 趣味指数：★★★★☆

实例说明

本实例在实现管理员登录功能的基础上，增加对管理员数据的更新和删除功能。可以对管理员的名称、密码和注册时间进行更新操作，同时也可以删除指定的管理员数据，其运行结果如图 12.15 所示。



图 12.15　管理员数据的更新和删除

关键技术

这里仍应用 ADODB 连接、操作 Oracle 数据库。以数据表中数据的 ID 为条件，执行 update 更新和 delete 删除语句。

在本实例中，管理员登录成功后，应用 SESSION 会话机制将登录的管理员名称存储到 SESSION 变量中，通过这个 SESSION 来控制用户的权限，如果是管理员则可以执行管理员数据的更新和删除操作，否则将给出提示信息，并跳转到管理员登录页面，其关键代码如下：

```
<?php
session_start();                                    //初始化 SESSION 变量
header ( "Content-type: text/html; charset=utf-8" );  //设置文件编码格式
if($_SESSION['name']!=""){                          //判断管理员的名称是否为空
include("conn/conn.php");                           //包含数据库连接文件
if($_POST['Submit2']=="删除"){
    $sql="delete from tb_admin where id='".$_POST['conn_id']."'";
    $rst = $conn->execute ( $sql ) or die ( 'error:   ' . $conn->errorMsg () );
    if($rst){
        echo "<script>alert('管理员删除成功！');window.location.href='main.php';</script>";
    }else{
```

```
            echo "<script>alert('管理员删除失败！ ');windwo.location.href='main.php'</script>";
        }
    }
if($_POST['Submit']=="更新"){
    $sql="update tb_admin set name='".iconv("utf-8","gb2312",$_POST['name'])."',pass='".md5(iconv("utf-8","gb2312",$_POST['pass']))."',
dates='".$_POST['dates']."' where id='".$_POST['conn_id']."'";
    $rst = $conn->execute ( $sql ) or die ( 'error：  ' . $conn->errorMsg () );
    if($rst){
            echo "<script>alert('管理员更新成功！ ');window.location.href='main.php';</script>";
    }else{
            echo "<script>alert('管理员更新失败！ ');windwo.location.href='main.php'</script>";
    }
}
}else{
 echo "<script>alert('您不具备管理员权限！ ');windwo.location.href='index.php'</script>";
}
?>
```

其中，在管理员登录的处理页 index_ok.php 中，首先初始化 SESSION 变量。在通过登录验证后，将登录用户名存储到 SESSION 变量中。

在数据的更新和删除处理页中，首先初始化 SESSION 变量，然后，通过判断 SESSION 变量的值是否为空来决定当前用户是否具有管理员的权限。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 Oracle 数据库的连接。

（2）创建 index.php 文件，添加 form 表单，将管理员登录的数据提交到 index_ok.php 文件中。

（3）创建 index_ok.php 文件，完成登录用户的验证操作。首先，初始化 SESSION 变量，包含数据库连接文件。然后，执行 select 查询语句，根据 GetArray()函数返回的数组元素个数判断登录用户的用户名和密码是否正确，如果正确，则将用户名赋给 SESSION 变量，并且跳转到 main.php 页面；否则，给出提示信息，跳转到 index.php 页面。

（4）创建 main.php 文件，在 form 表单中循环输出管理员数据，并且创建管理员数据的更新和删除按钮，将数据提交到 main_ok.php 文件完成更新和删除操作，其关键代码如下：

```
<?php
include_once 'conn/conn.php';                                        //载入数据库链接文件
$sqlstr = 'select * from tb_admin ';                                 //SQL 查询语句
$rst = $conn->execute ( $sqlstr ) or die ( 'error：  ' . $conn->errorMsg () );  //执行查询语句
while ( ! $rst->EOF ) {                                              //wihle 语句循环输出结果
?>
<form id="form<?php echo $rst->fields ['0']; ?>" name="form<?php echo $rst->fields ['0']; ?>" method="post" action="main_ok.php">
<tr>
<td align="center" bgcolor="#FFFFFF">
            <input name="name" type="text" id="name" size="15" value="<?php echo iconv("gb2312","utf-8",$rst->fields ['1']);?>" /></td>
      <td align="center" bgcolor="#FFFFFF">
            <input name="pass" type="password" id="pass" value="<?php echo iconv("gb2312","utf-8",$rst->fields ['2']);?>" size="15" /></td>
      <td align="center" bgcolor="#FFFFFF">
            <input name="dates" type="text" id="dates" size="15" value="<?php echo iconv("gb2312","utf-8",$rst->fields ['3']);?>" /></td>
      <td align="center" bgcolor="#FFFFFF">
            <input type="hidden" name="conn_id" value="<?php echo $rst->fields ['0']; ?>" />
            <input type="submit" name="Submit" value="更新" />  
<input type="submit" name="Submit2" value="删除" />
</td>
</tr>
</form>
<?php
        $rst->movenext ();                                           //指针下移
}
$rst->close ();                                                      //关闭连接
$conn->close ();
?>
```

（5）创建 main_ok.php 文件，完成管理员数据的更新或者删除操作。首先，初始化 SESSION 变量，通过

SESSION 变量判断当前用户是否具有操作权限，如果有权限，则可以根据按钮的值执行更新或者删除操作；否则将给出提示信息，并跳转到管理员登录页面，其代码如下：

```
<?php
session_start();
header ( "Content-type: text/html; charset=utf-8" );                //设置文件编码格式
if($_SESSION['name']!=""){
include("conn/conn.php");
if($_POST['Submit2']=="删除"){
 $sql="delete from tb_admin where id='".$_POST['conn_id']."'";
 $rst = $conn->execute ( $sql ) or die ( 'error：  ' . $conn->errorMsg () );
 if($rst){
        echo "<script>alert('管理员删除成功！');window.location.href='main.php';</script>";
 }else{
        echo "<script>alert('管理员删除失败！');windwo.location.href='main.php'</script>";
 }
}
if($_POST['Submit']=="更新"){
 $sql="update tb_admin set
name='".iconv("utf-8","gb2312",$_POST['name'])."',pass='".md5(iconv("utf-8","gb2312",$_POST['pass']))."',dates='".$_POST['dates']."' where
id='".$_POST['conn_id']."'";
 $rst = $conn->execute ( $sql ) or die ( 'error：  ' . $conn->errorMsg () );
 if($rst){
        echo "<script>alert('管理员更新成功！');window.location.href='main.php';</script>";
 }else{
        echo "<script>alert('管理员更新失败！');windwo.location.href='main.php'</script>";
 }
}
}else{
 echo "<script>alert('您不具备管理员权限！');windwo.location.href='index.php'</script>";
}
?>
```

秘笈心法

心法领悟 533：SESSION 变量的应用。

通过 SESSION 变量的值来判断当前用户的权限是非常实用的方法，其原理是：在会员或者管理员登录成功后，将登录的名称或者一个指定的值赋给 SESSION 变量，由于 SESSION 变量可以实现在不同页面之间的传递，所以可以在具有访问权限限制的页面中判断这个 SESSION 变量的值，如果变量值为真，则说明这个用户具有访问权限，否则说明当前用户不具备访问权限。

实例 534　Smarty+Adodb 完成 Oracle 数据的分页显示

光盘位置：光盘\MR\12\534

高级

趣味指数：★★★★☆

实例说明

在本实例中，同样应用 ADODB 连接、操作 Oracle 数据库，其不同之处是将连接、操作数据库的方法封装到类中，同时将数据库中数据的分页方法也封装到类中，还有通过 Smarty 模板完成页面的动静分离。本实例完成数据库中员工信息的分页输出，其运行结果如图 12.16 所示。

ID	名称	职务	年龄	入职时间
5	欧阳	部门经理	28	2010-08-20
6	东方	设计师	28	2010-08-20
8	明日科技	总经理	28	2008-08-10

共有员工 8 名 每页显示 3 名 第 2 页共 3 页　首页 上一页 下一页 尾页

图 12.16　Smarty+Adodb 完成 Oracle 数据的分页显示

关键技术

本实例的关键是应用面向对象技术，将 Oracle 数据库的连接、操作以及分页方法都封装到类中，同时还包括 Smarty 模板的配置方法。应用类的实例化返回的对象调用类中方法，完成与 Oracle 数据库的连接、操作和分页，最终通过 Smarty 模板完成页面的动静分离。

（1）Oracle 数据库连接类 ConnDB，不但可以连接 Oracle 数据库，还可以连接 MySQL、SQL Server 和 Access 数据库。该类存储于 system\ system.smarty.inc.php 文件中，其内容如下：

```
class ConnDB{
	var $dbtype;
	var $host;
	var $user;
	var $pwd;
	var $dbname;
	var $debug;
	var $conn;
	function ConnDB($dbtype,$host,$user,$pwd,$dbname,$debug=false){			//构造方法，为成员变量赋值
		$this->dbtype=$dbtype;
		$this->host=$host;
		$this->user=$user;
		$this->pwd=$pwd;
		$this->dbname=$dbname;
		$this->debug=$debug;
	}
	function GetConnId(){							//实现与不同数据库的连接并返回连接对象
		require("../adodb5/adodb.inc.php");				//调用 ADODB 类库文件
		if($this->dbtype=="oci8" || $this->dbtype=="mysql" || $this->dbtype=="mssql"){	//判断成员变量传递的数据库类型
			if($this->dbtype=="mysql"){				//判断如果是 MySQL 数据库
				$this->conn=NewADOConnection("mysql");		//执行与 MySQl 数据库的连接
			}else if($this->dbtype=="mssql"){
				$this->conn=NewADOConnection("mssql");
			$this->conn->Connect($this->host,$this->user,$this->pwd,$this->dbname);	//数据库连接的用户、密码
			}else{
				$this->conn = ADONewConnection("oci8");		//建立一个 oracle 连接
				$this->conn->Connect($this->dbname,$this->user,$this->pwd);
			}
		}elseif($this->dbtype=="access"){				//判断如果使用的是 Access 数据库
			$this->conn=NewADOConnection("access");
			$this->conn->Connect("Driver={Microsoft Access Driver
(*.mdb)};Dbq=".$this->dbname.";Uid=".$this->user.";Pwd=".$this->pwd.";");	//执行连接 Access 数据库
		}
		$this->conn->Execute("set names utf8");			//设置数据库的编码格式
		if($this->dbtype=="mysql")
			$this->conn->debug=$this->debug;
		return $this->conn;						//返回连接对象
	}
	function CloseConnId(){						//定义关闭数据库的方法
		$this->conn->Disconnect();				//执行关闭的操作
	}
}
```

（2）数据库操作类 AdminDB，执行对 Oracle 数据库的操作语句，如果是 select 查询语句，成功则返回结果集数组，否则返回 FALSE；如果是 update、insert 或者 delete 删除语句，成功则返回 TRUE，否则返回 FALSE。该类存储于 system\ system.smarty.inc.php 文件中，其内容如下：

```
class AdminDB{
	function ExecSQL($sqlstr,$conn){				//定义方法，参数为 SQl 语句和连接数据库返回的对象
		$sqltype=strtolower(substr(trim($sqlstr),0,6));		//截取 SQL 中的前 6 个字符串，并转换成小写
		$rs=$conn->Execute($sqlstr);				//执行 SQL 语句
		if($sqltype=="select"){					//判断如果 SQL 语句的类型为 select
			$array=$rs->GetRows();				//执行该语句，获取查询结果
			if(count($array)==0 || $rs==false)		//判断语句是否执行成功
				return false;				//如果查询结果为 0，或者执行失败，则返回 false
			else
```

```
                    return $array;                                          //否则返回查询结果的数组
            }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){
                //判断如果 SQL 语句类型不为 select，则执行如下操作
                if($rs)
                    return true;                                            //执行成功返回 TRUE
                else
                    return false;                                           //是否返回 FALSE
            }
        }
}
```

（3）分页类 SepPage，定义两个方法 ShowDate()和 ShowPage()完成数据的分页输出。该类存储于 system\system.smarty.inc.php 文件中，代码请参考本书光盘。

（4）Smarty 模板的配置类 SmartyProject，完成 Smarty 应用的配置操作，指定模板页存储位置，指定编译文件存储位置。该类存储于 system\ system.smarty.inc.php 文件中，其内容如下：

```
require("../Smarty/Smarty.class.php");                              //调用 Smarty 文件
class SmartyProject extends   Smarty{                               //定义类，继承 Smarty 父类
 function SmartyProject(){                                          //定义方法，配置 Smarty 模板
        $this->template_dir = "./";                                 //指定模板文件存储在根目录下
        $this->compile_dir = "./system/Smarty/templates_c/";        //指定编译文件存储位置
        $this->config_dir = "./system/Smarty/configs/";
        $this->cache_dir = "./system/Smarty/cache/";
     }
}
```

设计过程

（1）创建 system 文件夹。首先，定义 system.smarty.inc.php 文件，封装 Oracle 数据库的连接和操作方法，以及 Smarty 的配置方法。然后，定义 system.inc.php 文件，对数据库的连接和操作类以及 Smarty 配置类进行实例化，并返回连接对象。最后，创建 Smarty 文件夹，定义 Smarty 的编译文件、配置文件和缓存文件的存储目录。

（2）创建 index.php 文件。首先，包含类的实例化文件。然后，应用 Smarty 生成一个注册函数，实现对数据库中数据的编码格式转换。接着，调用分页类中的方法完成数据的分页输出，并将返回的结果存储到模板变量中。最后，指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");                              //包含配置文件
function unHtml($params) {                                          //创建模板函数
extract($params);                                                   //读取数据
    $str=iconv("gb2312","utf-8",$text);
    return $str;                                                    //返回截取结果
}
$smarty->register_function("Util", "unHtml");                       //注册模板函数
$arr=$seppage->ShowDate("select * from tb_user ",$conn,3,$_GET["page"]);   //调用分页类，实现分页功能
if(!$arr){
      $smarty->assign("isbbs","F");
}else{
        $smarty->assign("isbbs","T");
 $smarty->assign("showpage",$seppage->ShowPage("员工","名","","a1"));      //定义输出分页数据的模板变量 showpage
        $smarty->assign("arr",$arr);
}
 $smarty->assign('title','Smarty+Adodb 完成 ORACL 数据库中数据的分页显示');
 $smarty->display('index.html');
 ?>
```

（3）创建 index.html 模板页，应用 section 语句循环输出数据库中的数据，并且应用注册模板函数对数据的编码格式进行转换，同时输出模板变量中存储的分页超链接，其关键代码如下：

```
 {if $isbbs=="T"}
        {section name=arrid loop=$arr}
<tr>
            <td align="center" bgcolor="#FFFFFF">{$arr[arrid].0}</td>
            <td align="center" bgcolor="#FFFFFF">{Util text=$arr[arrid].1}</td>
            <td align="center" bgcolor="#FFFFFF">{Util text=$arr[arrid].2}</td>
            <td align="center" bgcolor="#FFFFFF">{Util text=$arr[arrid].3}</td>
```

```
                <td align="center" bgcolor="#FFFFFF">{Util text=$arr[arrid].4}</td>
</tr>
        {/section}
{/if}
{$showpage}
```

秘笈心法

心法领悟 534：ADODB 类库和 Smarty 模板。

有关 ADODB 类库的具体应用将在第 13 章中进行详细讲解，而 Smarty 模板的应用可以参考本书第 14 章。在本实例中，封装数据库连接、操作类、分页类和 Smarty 模板的配置类是一个非常不错的方法，推荐读者使用，并且可以对它进行不断地完善。例如，扩展其支持更多的数据库、更改分页类的方法以及更改 Smarty 模板中配置文件的存储路径等。

实例 535	Oracle——存储过程更新员工信息 光盘位置：光盘\MR\12\535	高级 趣味指数：★★★★☆

实例说明

在本实例中，应用 Oracle 数据库的存储过程技术，实现对员工信息表中数据的更新操作，其运行结果如图 12.17 所示。

图 12.17 存储过程更新员工信息

关键技术

存储过程是为执行一定的任务而组合在以前的 SQL 和/或 PL/SQL 语句集，通常由以下几部分组成：声明部分、执行部分和异常处理部分，其中执行语句是必须的，而其他两部分则是可选的。

创建存储过程的前提条件是用户具有创建过程的权限，这个权限可以由 DBA 赋予用户，其语法如下：

```
GRANT CREATE ANY PROCEDURE TO UserName;
```

取消权限的语法如下：

```
REVOKE CREATE ANY PROCEDURE FROM UserName;
```

Oracle 数据库中创建存储过程的语法如下：

```
CREATE[OR REPLACE] PROCEDURE procedure_name(
 argument1[{in|out|in out}] type,
 argument2[{in|out|in out}] type,
 …
 argument2[{in|out|in out}] type
)
{is | as}
<!--类型变量的说明-->
BEGIN
<!--执行部分-->
```

```
[EXCEPTION]
<!--可选的异常处理部分-->
END[procedure_name];
```

参数说明：

replace：关键字为可选值，多数情况下会使用。其作用为创建存储过程时，如果同名过程已经存在，则删除同名过程，然后进行创建操作。有一点需要注意，同一用户下的过程是唯一标识的，如果创建时不使用 replace 关键字，当出现同名过程时，会给出 ORA-00955 错误，表示名称已经被现有对象使用。

procedure_name：创建过程的名称，它在数据库中同一用户下是唯一标识。

argument：过程的参数名称，为可选参数。

in、out、in out：参数的模式，默认的模式为 in。in 模式为输入参数模式，具备只读属性，不可以在执行过程中赋值；out 模式为输出参数模式；in out 为输入/输出参数模式。

type：指定参数的数据类型。其对形参的定义只可以指定类型，不可以限定精度。

is|as：该关键字到 BEGIN 关键字之间为过程的声明部分。声明包括类型、游标、常量、变量、异常和嵌套子程序的声明，此处不需 DECLARE 关键字，而且此声明部分中声明的都是本地和局部声明，即过程结束时声明终止。

BEGIN 关键字和 END 关键字之间为过程的执行部分，如果过程中存在异常处理，则 BEGIN 和 EXCEPTION 关键字之间为过程的执行部分，EXCEPTION 和 END 关键字之间为异常处理部分。

在本实例中创建的存储过程如下：

```
CREATE OR REPLACE PROCEDURE PROC_USER (
  p_id IN integer,
  p_position IN VARCHAR
) AS
BEGIN
  update tb_user set position=p_position where id=p_id;
 commit;
END PROC_USER;
```

设计过程

（1）创建 conn 文件，编写 conn.php 文件，完成与 Oracle 数据库的连接。

（2）创建 index.php 文件，包含数据库连接文件，应用 Oracle 函数读取 Oracle 数据库中的数据，并通过 while 语句完成数据的循环输出。创建 form 表单，完成员工更新信息的提交，将更新数据提交到 index_ok.php 文件中，其关键代码如下：

```
<?php
include("conn/conn.php");
$query="select * from tb_user ";
$result=oci_parse($conn,$query);
$r = oci_execute($result, OCI_DEFAULT);
while($row=oci_fetch_array($result,OCI_RETURN_NULLS)){
?>
<form id="form<?php echo $row[0];?>" name="form<?php echo $row[0];?>" method="post" action="update_ok.php">
<tr>
 <td align="center" bgcolor="#FFFFFF"><?php echo $row[0];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $row[1];?></td>
      <td align="center" bgcolor="#FFFFFF">
         <input name="position" type="text" value="<?php echo $row[2];?>" size="15" />
            <input type="hidden" name="conn_id" value="<?php echo $row[0];?>" /></td>
         <td align="center" bgcolor="#FFFFFF"><?php echo $row[3];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $row[4];?></td>
      <td align="center" bgcolor="#FFFFFF"><input type="submit" name="Submit" value="更新" /></td>
</tr>
</form>
<?php
}
oci_close($conn);
?>
```

（3）创建 index_ok.php 文件，获取表单中提交的数据，调用存储过程完成员工信息的更新操作，其关键代

码如下：

```
<?php
header ( "Content-type: text/html; charset=utf-8" );                //设置文件编码格式
include("conn/conn.php");
$sql = "BEGIN PROC_USER(:id, :position); END;";                     //定义 SQL 语句
$stmt = OCIParse($conn, $sql);                                      //执行 SQL 语句
//执行绑定
OCIBindByName($stmt, ":id", $_POST['conn_id'], 4);                  //参数说明：绑定 php 变量 conn_id 到位置:id，并设定绑定长度为 4 位
OCIBindByName($stmt, ":position", $_POST['position'], 50);
if(OCIExecute($stmt)){                                              //调用并执行存储过程
 echo "<script>alert('员工信息更新成功!');window.location.href='index.php';</script>";
}else{
        echo "<script>alert('员工信息更新失败!');window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 535：在 PHP 动态页中调用存储过程。

在 PHP 动态页中调用存储过程。首先，定义 SQL 语句，执行存储过程中的更新语句。然后，通过 ociparse()函数配置 Oracle 语句预备执行。接着，通过 OCIBindByName()函数绑定一个 PHP 变量到一个 Oracle 标识符。最后，通过 OCIExecute()函数执行 Oracle 语句，完成员工信息的更新操作。

12.4 Oracle 数据的导入和导出

本节将介绍 Oracle 数据库中数据的导入、导出方法。

实例 536	Oracle 数据库中数据导出到文本文件 光盘位置：光盘\MR\12\536	高级 趣味指数：★★★★☆

实例说明

本实例将 Oracle 数据库中的数据表导出到本地磁盘 C 下的 mrsoft 文件夹中，并将导出的文本文件命名为 oracle.dmp。导出的结果如图 12.18 所示。



图 12.18 导出的文本文件

关键技术

本实例应用 Oracle 数据库中数据库逻辑备份工具 Export。利用 Export 工具可以在数据库打开状态下备份数据库。Export 把数据库中的对象导出到一个二进制文件中。该工具也是数据库间进行迁移的一个工具。

Export 有 3 种方法：行命令法、交互式方法和参数文件方法。下面主要介绍本实例中使用的参数文件方法。

将 EXP 命令的各个参数放在一个参数文件中，参数文件的命令格式如下：

```
EXP [用户名| 口令] PARFILE = 参数文件路径及名称
```

以下是一个参数文件的示例，代码如下：

```
FULL = Y
FILE = expfile.dmp
GRANTS= Y
INDEXES= Y
```

其中，FULL 设置为 Y 时，会导出除 SYS 方案对象外的所有的方案对象，如果设置为 FULL，要求用户有 EXP_FULL_DATABASE 权限。

FILE 参数用来设置导出文件的目录和文件名。

GRANTS 参数默认值为 Y，用来设置是否导出权限信息。

INDEXES 参数默认值为 Y，用来设置是否导出表和聚集的索引。

设计过程

（1）选择“开始”/“程序”/“附件”/“命令提示符”命令，进入 DOS 命令环境。

（2）在 DOS 提示符下输入“EXP 用户名|口令命令”（EXP 不区分大小写），进入导出界面，代码如下：

```
exp system/mrsoft tables=(tb_user,tb_admin) file=C:\mrsoft\oracle.dmp
```

（3）按 Enter 键，执行本实例，其运行结果如图 12.19 所示。



图 12.19　Oracle 数据库中数据表的导出

秘笈心法

心法领悟 536：Export 工具导出表空间和导出数据库。

通过 Export 工具不但可以导出表，还可以导出表空间、分区和数据库。

（1）导出表空间的语法如下：

```
exp /demo/demo tablespaces=data
```

（2）导出数据库，将导出所有的数据库对象及其数据，并存储到指定的文件系统中。

① 完全导出。将所有的数据库对象及数据导出到文件中，并为积累导出和增量导出的基点安排一个较长的周期进行完全导出，其命令如下：

```
exp system/manager full=y inctype=complete file=export.dmp
```

② 增量导出。将上次导出以来已经发生变化的对象结构及数据导出到文件中，其命令如下：

```
exp system/manager full=y inctype=incremental file=export.dmp
```

③ 积累导出。将上次进行积累或完全导出以来已经发生变更的对象及其数据库导出到文件中，其命令如下：

```
exp system/manager full=y inctype=cumulative file=export.dmp
```

实例 537	文本文件中数据导入到 Oracle 光盘位置：光盘\MR\12\537	高级 趣味指数：★★★★☆

实例说明

在本实例中，讲解将文本文件中数据导入到 Oracle 数据库中。数据的导入应用的是 import 工具，同样也分为导入表、导入方案和导入数据库 3 种。在本实例中将文本文件 oracle.dmp 中存储的数据导入到 Oracle 数据库中。

说明：在执行将文本文件中数据导入到 Oracle 数据库时，必须确保在 Oracle 数据库中不存在与文本文件中同名的数据表，否则导入操作不会成功。





第13章 ADODB 类库

- 封装 ADODB 操作类
- ADODB 的应用
- ADODB 中的错误处理
- ADODB 操作数据库

13.1　封装 ADODB 操作类

ADODB 是 Active Data Objects Data Base 的简称，是一种 PHP 存取数据库的中间函式组件。本节将介绍如何对 ADODB 操作数据库的方法进行封装，从而使通过 ADODB 操作数据库的方法更加简单。

说明： 因为 ADODB 属于第三方组件，所以在本书的光盘中没有提供 adodb 类库文件，需要读者自行下载，并复制到本章的 13 文件夹下，将其命名为 adodb5，然后才可以运行本章的实例。

实例 538	ADODB 连接数据库类 光盘位置：光盘\MR\13\538	高级 趣味指数：★★★★☆

实例说明

连接数据库类的封装，意味着通过 ADODB 连接不同的数据库的方法更加简单、更容易更新和修改。只要向类中传递对应的参数值，即可完成与不同数据库服务器的连接。在本实例中，以 MySQL 数据库服务器为例，实现与 db_database13 数据库的连接，连接成功后弹出提示信息，其运行结果如图 13.1 所示。



图 13.1　ADODB 连接数据库类

关键技术

ADODB 连接数据库系统使用 ADONewConnection()函数；如果选择持久化连接则使用 PConnect()函数，如果选择非持久化连接则使用 Connect()函数。

（1）ADONewConnection()函数，连接数据库系统，其语法如下：

```
ADONewConnection($databaseType)
```

参数$databaseType 表示要连接的数据库系统的名称，例如 mysql、mssql 等。

（2）PConnect()函数，实现与数据库的持久化连接，其语法如下：

```
PConnect($host,[$user],[$password],[$database])
```

参数说明：

$host：数据库系统的服务器所在地址。如果是本机操作，参数值为 localhost。

$user：数据库用户名。

$password：数据库密码。

$database：使用到的数据库。

（3）Connect()函数，实现与数据库的非持久化连接，其语法如下：

```
Connect($host,[$user],[$password],[$database])
```

参数说明：

$host：数据库系统的服务器所在地址。如果是本机操作，参数值为 localhost。

$user：数据库用户名。

$password：数据库密码。

$database：使用到的数据库。

掌握连接数据库服务器所使用的函数之后，就是定义类、方法和参数，完成与指定数据库服务器的连接，并返回连接标识，其具体代码将在设计过程中进行介绍。

设计过程

（1）创建 found.database.php 文件，定义 ConnDB 连接数据库类，定义 ConnDB()方法为成员变量赋值；定义 GetConnId()方法实现与不同数据库服务器的连接，并返回连接标识；最后定义 CloseConnId()方法关闭数据库连接，其关键代码如下：

```
<?php
//数据库连接类
class ConnDB{
	var $dbtype;										//数据库服务器类型
	var $host;										//服务器名称
	var $user;										//用户名
	var $pwd;										//密码
	var $dbname;										//数据库名称
	var $debug;										//是否返回执行的 SQL 语句，默认设置为返回 SQL
	var $conn;										//连接数据库时返回的连接标识
	function ConnDB($dbtype,$host,$user,$pwd,$dbname,$debug=true){	//构造方法，为成员变量赋值
		$this->dbtype=$dbtype;
		$this->host=$host;
		$this->user=$user;
		$this->pwd=$pwd;
		$this->dbname=$dbname;
		$this->debug=$debug;
	}
	function GetConnId(){										//实现与不同数据库的连接并返回连接对象
		require("../adodb5/adodb.inc.php");							//调用 ADODB 类库文件
			if($this->dbtype=="mysql" || $this->dbtype=="mssql"){			//判断成员变量传递的数据库类型
				if($this->dbtype=="mysql")						//判断如果是 MySQL 数据库
					$this->conn=NewADOConnection("mysql");			//执行与 MySQl 数据库的连接
				else
					$this->conn=NewADOConnection("mssql");
				$this->conn->Connect($this->host,$this->user,$this->pwd,$this->dbname);	//数据库连接的用户、密码
			}elseif($this->dbtype=="access"){						//判断如果使用的是 Access 数据库
				$this->conn=NewADOConnection("access");
				$this->conn->Connect("Driver={Microsoft Access Driver (*.mdb)};Dbq=".$this->dbname.";Uid=".$this->user.";
Pwd=".$this->pwd.";");										//执行连接 Access 数据库
			}
			$this->conn->Execute("set names utf8");					//设置数据库的编码格式
			if($this->dbtype=="mysql")
				$this->conn->debug=$this->debug;
				return $this->conn;							//返回连接对象
	}
	function CloseConnId(){										//定义关闭数据库的方法
		$this->conn->Disconnect();								//执行关闭的操作
	}
}
?>
```

（2）创建 connect.database.php 文件，实例化数据库连接类 ConnDB，并向类中传递对应的参数，完成与 MySQL 数据库服务器中 db_database13 数据库的连接，其关键代码如下：

```
<?php
require("found.database.php");									//包含类文件
$connobj=new ConnDB("mysql","localhost","root","111","db_database13",false);		//数据库连接类实例化
$conn=$connobj->GetConnId();									//返回连接标识
?>
```

（3）创建 index.php 文件，包含数据库连接类的实例化文件，根据返回的连接标识判断数据库是否连接成功，其关键代码如下：

```
<?php
include_once 'conn/connect.database.php';							//包含数据库连接类的实例化文件
if($conn){
 echo "<script>alert('数据库连接成功！');window.location.href='http://www.mrbccd.com';</script>";
```

```
}else{
echo "<script>alert('数据库连接失败！');window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 538：通过 ADODB 连接数据库服务器必须注意的问题。

定义连接数据库类，再定义连接方法 GetConnId()时，必须先通过 require()语句包含 ADODB 类库文件 adodb.inc.php，否则其他一切操作都无从谈起。

建议在程序的开发阶段运用连接数据库操作类时，将参数$debug 的值设置为 TRUE，这样可以返回执行的 SQL 语句，便于获取 SQL 语句在执行过程中出现的错误。如果是在生产环境中运用连接数据库类，那么就要将$debug 的值设置为 FALSE。

实例 539　ADODB 操作数据库类

光盘位置：光盘\MR\13\539

高级

趣味指数：★★★★☆

实例说明

既然可以将 ADODB 连接数据库的方法封装到类中，那么同样也可以将 ADODB 操作数据库的方法封装到类中。在本实例中，介绍如何将 ADODB 操作数据库的方法封装到类中，并且通过该方法读取 MySQL 数据库服务器的 db_database13 数据库中的 tb_bccd 数据表中的数据，其运行结果如图 13.2 所示。

ID	名称	版本	时间
1	VC编程词典	标准版	2010-07-29
2	VB编程词典	标准版	2010-07-29
3	JAVA编程词典	标准版	2010-07-29

图 13.2　ADODB 操作数据库类

关键技术

ADODB 操作数据库类为 AdminDB，在操作数据库的 ExecSQL()方法中，关键是以下几个函数的运用：

（1）strtolower()函数，将字符串转换为小写字母，其语法如下：

```
string strtolower(string str)
```

运用本函数的目的是将 SQL 语句中的字符串都转换为小写。

（2）substr()函数，从指定的字符串中按照指定的位置截取一定长度的字符，其语法如下：

```
string substr(string str,int start,int length)
```

参数说明：

str：指定截取的字符串。

start：指定开始截取字符串的位置，如果为负数，则从字符串的末尾开始截取。

length：可选参数，指定截取字符串的长度。如果为负数，则表示取倒数第 length 个字符。

注意：本函数中参数 start 的指定位置是从 0 开始计算的，即字符串中的第一个字符表示为 0。

运用 substr()函数对 SQL 语句中前 6 个字符串进行截取，其目的是获取 SQL 语句的类型（insert、select 或者 update），判断其执行的是什么操作。

（3）trim()函数，删除字符串中首尾的空白或者其他字符。

（4）execute()函数，ADODB 中执行 SQL 语句的方法，并返回一个结果集（ADORecordSet 对象），失败则返回 FALSE，其语法如下：

```
execute($sql[,$inputarr=false])
```

参数说明：

$sql：指定要执行的 SQL 语句。

$inputarr：设置传入的结合变量，如果没有设置$inputarr，则$sql 为普通的 SQL 语句，函数 execute()的格式如下：

```
$connect->execute('select * from tb_user where id = 1')
```

如果设置了$inputarr，函数 execute()的格式如下：

```
$connect -> execute('select * from tb_user where id = ?',array($vl))
```

（5）GetRows()函数，是 GetArray() 的同义函数，是为了与 Microsoft ADO 相容定义的，其语法如下：

```
GetRows ([$number_of_rows])
```

返回从当前指针指向的记录开始，到（$number_of_rows – 1）行的全部记录的数组。参数$number_of_rows 表示指定的记录行，如果没有给出，则一直到 EOF 才停止。

设计过程

（1）创建 found.database.php 文件，定义数据库操作类 AdminDB，定义 ExecSQL()方法，根据传递的 SQL 语句执行对数据库的操作。在 ExecSQL 方法中，应用 substr()函数获取参数传递的 SQL 语句的类型，然后根据 SQL 语句的类型进行判断，如果是 select 查询语句，则执行 GetRow()方法，获取查询结果；如果是 update、insert 或者 delete 语句，则直接返回 TRUE 或者 FALSE，其代码如下：

```
class AdminDB{
 function ExecSQL($sqlstr,$conn){                                        //定义方法，参数为 SQL 语句和连接数据库返回的对象
        $sqltype=strtolower(substr(trim($sqlstr),0,6));                  //截取 SQL 中的前 6 个字符串，并转换成小写
        $rs=$conn->Execute($sqlstr);                                     //执行 SQL 语句
        if($sqltype=="select"){                                          //判断如果 SQL 语句的类型为 select
                $array=$rs->GetRows();                                   //执行该语句，获取查询结果
                if(count($array)==0 || $rs==false)                       //判断语句是否执行成功
                        return false;                                    //如果查询结果为 0，或者执行失败，则返回 FALSE
                else
                        return $array;                                   //否则返回查询结果的数组
        }elseif ($sqltype=="update" || $sqltype=="insert" || $sqltype=="delete"){
                //判断如果 SQL 语句类型不为 select、则执行如下操作
                if($rs)
                      return true;                                       //执行成功返回 TRUE
                else
                      return false;                                      //否则返回 FALSE
        }
 }
}
```

（2）创建 connect.database.php 文件，实例化数据库连接类 ConnDB 和数据库操作类 AdminDB，其关键代码如下：

```
<?php
require("found.database.php");                                              //包含类文件
$connobj=new ConnDB("mysql","localhost","root","111","db_database13",false);  //实例化数据库连接类
$conn=$connobj->GetConnId();                                                //返回连接标识
$admindb=new AdminDB();                                                     //实例化数据库操作类
?>
```

（3）创建 index.php 文件，包含数据库连接类和数据库操作类的实例化文件，定义 SQL 语句，调用数据库操作类中的 ExecSQL()方法执行 SQL 查询语句，最后通过 for 语句循环输出 ExecSQL()方法返回的查询结果，其关键代码如下：

```
<?php
include_once 'conn/connect.database.php';                     //包含数据库连接类和操作类的实例化文件
$sqlstr = 'select * from tb_bccd';                           //定义 SQL 语句
$array = $admindb->ExecSQL($sqlstr,$conn);                   //调用数据库操作类中的方法，执行查询操作
for($i=0;$i<count($array);$i++){                             //通过 for 循环输出数据库中的数据
```

```
?>
  <tr>
    <td align="center" bgcolor="#FFFFFF"><?php echo $array[$i][0];?></td>
    <td align="center" bgcolor="#FFFFFF"><?php echo $array[$i][1];?></td>
    <td height="22" align="center" bgcolor="#FFFFFF"><?php echo $array[$i][2];?></td>
    <td align="center" bgcolor="#FFFFFF"><?php echo $array[$i][3];?></td>
  </tr>
  <?php
  }
?>
```

秘笈心法

心法领悟 539：判断 SQL 语句类型的技巧。

SQL 语句的类型分为查询（select）、添加（insert）、更新（update）和删除（delete）4 种。这 4 个英文单词有一个共同的特点，即都是由 6 个英文字母组成。我们就是根据这个共同点对 SQL 语句的类型进行判断的。这也就是在实例中应用 substr()函数对 SQL 语句的前 6 个字符串进行截取的原因。

实例 540	ADODB 分页类 光盘位置：光盘\MR\13\540	高级 趣味指数：★★★★

实例说明

为了更方便地通过 ADODB 操作数据库中的数据，这里对 ADODB 类库中提供的分页函数进行整合，封装为一个 ADODB 分页类，并且运用 ADODB 分页类输出 MySQL 数据库 db_database13 中 tb_user 数据表中的数据，其运行结果如图 13.3 所示。

图 13.3　ADODB 分页类的应用

关键技术

在 ADODB 分页类 SepPage()中，主要整合下面几个 ADODB 中的函数：

（1）PageExecute()函数，分页功能函数，其语法如下：

```
PageExecute($sql, $nrows, $page);
```

参数说明：

$sql：SQL 查询语句。

$nrows：每页显示的记录数。

$page：保存当前页数，默认为 1。

（2）AbsolutePage()函数，返回当前页的页数，其语法如下：

```
AbsolutePage($page=-1)
```

该函数要与 PageExecute()函数配合使用。

（3）AtFirstPage()函数，如果当前页是第一页，则返回 TRUE，其语法如下：

```
AtFirstPage($status='')
```

要与 PageExecute()函数配合使用，判断当前是否是第一页。

（4）AtLastPage()函数，判断当前页是否是最后一页，其语法如下：

```
AtLastPage($status='');
```

同样要与 PageExecute()函数配合使用，判断当前页是否是最后一页，如果是则返回 TRUE。

本实例中就是整合了上述几个 ADODB 类库中的函数，封装了一个 ADODB 分页类。

设计过程

（1）创建 found.database.php 文件，定义 SepPage 分页类，定义 ShowData()方法，读取数据库中存储的数据，并且将查询结果返回到数组中。定义 ShowPage()方法，创建分页超链接，完成从数据库中读取数据的分页输出。SepPage 分页类的关键代码如下：

```
class SepPage{
 var $rs;
 var $pagesize;
 var $nowpage;
 var $array;
 var $conn;
 var $sqlstr;
 function ShowData($sqlstr,$conn,$pagesize,$nowpage){                    //定义方法
        if(!isset($nowpage) || $nowpage=="")                            //判断变量值是否为空
               $this->nowpage=1;                                        //定义每页起始页
        else
               $this->nowpage=$nowpage;
        $this->pagesize=$pagesize;                                      //定义每页输出的记录数
        $this->conn=$conn;                                              //连接数据库返回的标识
        $this->sqlstr=$sqlstr;                                          //执行的查询语句
        $this->rs=$this->conn->PageExecute($this->sqlstr,$this->pagesize,$this->nowpage);
        @$this->array=$this->rs->GetRows();                             //获取记录数
               if(count($this->array)==0 || $this->rs==false)
                      return false;
               else
                      return $this->array;
 }
 //省略了 ShowPage()方法中的内容
}
```

（2）创建 connect.database.php 文件，实例化数据库连接类 ConnDB、数据库操作类 AdminDB 和分页类 SepPage，其关键代码如下：

```
<?php
require("found.database.php");                                          //包含类文件
$connobj=new ConnDB("mysql","localhost","root","111","db_database13",false);   //实例化数据库连接类
$conn=$connobj->GetConnId();
$admindb=new AdminDB();                                                 //实例化数据库操作类
$seppage=new SepPage();                                                 //实例化分页类
?>
```

（3）创建 index.php 文件，包含 connect.database.php 文件，调用分页类中的 ShowData()方法，完成数据的分页输出；调用分页类中的 ShowPage()方法输出分页的超链接，其关键代码如下：

```
<?php
include_once 'conn/connect.database.php';                               //调用类中方法
$array=$seppage->ShowData("select * from tb_user",$conn,10,$_GET["page"]);   //分页读取数据库中数据
for($i=0;$i<count($array);$i++){                                       //循环输出数据库中数据
?>
<tr>
      <td height="20" align="center" bgcolor="#FFFFFF"><?php echo $array[$i][0];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $array[$i][1];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $array[$i][2];?></td>
</tr>
<?php
}
?>
<tr>
      <td height="35" ><?php echo $seppage->ShowPage("用户","个","参数 1","参数 2","a1");?></td>
</tr>
```

秘笈心法

心法领悟 540：ADODB 分页类的运用技巧。

在运用 ADODB 分页类进行数据的分页输出时，向 ShowData()方法中传递 4 个参数：SQL 查询语句、数据库的连接标识、每页显示的记录数和当前页码；向 ShowPage()方法中传递 5 个参数：数据的类型、数据的单位、超链接的参数一、参数二和超链接样式。

13.2　ADODB 的应用

ADODB 类库非常庞大，仅一个 adodb.inc.php 文件就有 120KB。本节将详细讲解 ADODB 类库中的连接数据库函数、操作数据库函数、公用变量、公用函数、结果集处理函数和错误处理函数等在实例中的具体应用。

实例 541	ADODB 分页技术 光盘位置：光盘\MR\13\541	高级 趣味指数：★★★☆

实例说明

在实例 540 中介绍了如何将 ADODB 中的分页函数封装到类中，并通过分页类完成数据的循环输出操作。而在本实例中，则还原分页函数的本来面貌，通过面向过程的方法编写一个数据分页输出的实例，了解分页函数的具体应用。本实例的运行结果如图 13.4 所示。

ID	名称	类型	添加时间
4	C#编程词典	标准版	Thu 29, Jul 2010
5	PHP编程词典	标准版	Thu 29, Jul 2010
6	C++编程词典	标准版	Thu 29, Jul 2010

当前是第2页/一共是4页　首页 上一页 下一页 尾页

图 13.4　ADODB 分页函数的应用

关键技术

有关 ADODB 类库中分页函数的详细介绍可以参考实例 540 关键技术中的内容，这里不再赘述。在本实例中，引入 rs2html()函数，直接将 PageExecute()函数返回的查询结果以表格的形式输出。

rs2html()函数，返回一个 HTML 表格格式的结果集，其语法如下：

```
rs2html($rst,[$table_attributes], [$col_titles])
```

参数说明：

$rst：要返回的结果集。

table_attributes：对表格的参数及属性的设置。

col_title：对字段的重新命名。

注意：运用 rs2html()函数的前提是，必须载入 tohtml.ini.php 文件。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，载入 adodb.ini.php 文件。实现与 MySQL 数据库服务器中 db_database13 数据库的连接，并设置数据库的编码格式为 utf-8，其代码如下：

```
<?php
include_once ('../adodb5/adodb.inc.php');                    //载入（include）adodb.inc.php 文件
$conn = ADONewConnection('mysql');                           //建立连接
$conn -> PConnect('localhost','root','111','db_database13'); //连接数据库
$conn -> execute('set names utf8');                          //设置编码格式

?>
```

（2）编程 index.php 文件。首先，包含数据库连接文件 conn.php，然后载入 tohtml.ini.php 文件，最后，定义 SQL，应用 ADODB 类库中的分页函数完成数据的分页输出，并且应用 rs2html()函数直接将分页查询的数据定义到表格中，其关键代码如下：

```
<?php
        include_once 'conn/conn.php';                         //载入数据库连接文件
        include ('../adodb5/tohtml.inc.php');                 //载入 tohtml.inc.php 文件
        $sql = 'select * from tb_bccd';                       //查询语句
        $num = 3;                                             //每页显示的记录数
        if (isset ( $_GET ['n_page'] )) {                     //判断当前页码
                $c_page = $_GET ['n_page'];                   //将$n_page 赋给变量$c_apge
        } else {
                $c_page = 1;                                  //初始化变量$c_page
        }
        $rst = $conn->PageExecute ( $sql, $num, $c_page );    //执行 PageExecute()函数
        if (false != $rst) {
?>
    <tr>
        <td><?php
                rs2html ( $rst, 'width="775" border="1" align="center" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF"
bgcolor="#999999"', array ('ID', '名称', '类型', '添加时间' ) );
                ?></td>
//省略了部分代码
  <?php
        }
?>
```

秘笈心法

心法领悟 541：将分页读取的数据存储到缓存中。

PageExecute()函数可以执行分页功能，但是如果要将分页查询的结果存储到缓存中，就必须使用 CachePageExecute($sec, $sql, $nrows, $page)函数，其不但具备 PageExecute()函数的功能，而且可以将显示后的记录放到缓存中。在$sec 秒内，如果需要重复查看，将从缓存中读取数据。

实例 542　Adodb_pager 类实现分页功能

光盘位置：光盘\MR\13\542

高级

趣味指数：★★★★☆

实例说明

在 ADODB 类库中，不只提供了分页函数，还提供了封装好的分页类供开发者使用。在本实例中，运用 ADODB 类库中提供的 ADODB_pager 类实现分页功能，分页输出编程词典的注册用户，其运行结果如图 13.5 所示。

图 13.5　ADODB_pager 分页类

关键技术

ADODB_Pager()函数，是 ADODB_Pager 类的构造函数，通过类中的 render()函数可以实现分页功能，其语法如下：

```
ADODB_Pager($conn, $sql, $id = 'adodb', $showPageLinks = false)
```

参数$conn 是数据库连接对象；参数$sql 是执行的 SQL 语句；参数$id 为每个分页的 id 号；参数$showPageLinks 设置是否显示各个页的链接，默认为 FALSE。

提示： 应用 ADODB_pager()函数实现分页功能的前提条件，必须载入 adodb-pager.ini.php 文件。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，载入 adodb.ini.php 文件。实现与 MySQL 数据库服务器中 db_database13 数据库的连接，并设置数据库的编码格式为 utf-8，其代码如下：

```
<?php
include_once ('../adodb5/adodb.inc.php');                //载入（include）adodb.inc.php 文件
$conn = ADONewConnection('mysql');                        //建立连接
$conn -> PConnect('localhost','root','111','db_database13');   //连接数据库
$conn -> execute('set names utf8');                       //设置编码格式

?>
```

（2）创建 index.php 文件，完成数据的分页输出。首先，包含数据库连接文件。然后，载入 adodb-pager.ini.php 文件。最后，实例化 ADODB_Pager 类，调用 Render()方法，设置每页显示 10 条记录，其关键代码如下：

```
<?php
include 'conn/conn.php';                                  //载入数据库链接文件
include_once ' adodb-pager.inc.php';                      //载入 adodb_parper.ini.php 文件
/*   声明对象   */
$pager = new ADODB_Pager($conn,"select id,user as '用户名', pass as '密码',dates as '注册时间' from tb_user");
/*   调用$Render 函数确定每页显示多少记录   */
$pager -> Render(10);
?>
```

秘笈心法

心法领悟 542：修改 ADODB_Pager 类中定义的分页方法的样式。

修改 ADODB_Pager 类中定义的分页方法的样式，即对载入的 adodb-pager.ini.php 文件进行修改。可以修改的内容包括超链接、表格背景、宽度、边框宽度和颜色等。在本实例中，将超链接修改为中文，并且重新定义了表格的宽度。为了便于读者使用，还将修改后的 adodb-pager.ini.php 从 ADODB 类库中复制到实例的根目录下。

实例 543	ADODB 操作结果集的方法 光盘位置：光盘\MR\13\543	高级 趣味指数：★★★☆

实例说明

通过 ADODB 类库操作数据库，不但要掌握数据库的连接方法，还要掌握操作结果集的方法。在本实例中，将介绍 ADODB 类库中操作结果集的方法，并且应用其中的方法循环输出数据表中的字段和值，其运行结果如图 13.6 所示。



图 13.6　ADODB 操作结果集的方法

关键技术

当使用 execute()函数执行 SQL 指令时，会回传一个 ADORecordSet 对象。通过对 ADORecordSet 对象的控制，可以对结果集进行很多操作。操作结果集的方法如下：

（1）fields

该变量保存的是当前指针所指向的记录。

（2）EOF

该变量记录当前指针是否指向最后一条记录。如果是，则设定为 TRUE，否则为 FALSE。

（3）GetArray([$number_of_rows])

返回从当前指针指向的记录开始，到（$number_of_rows – 1）行的全部记录的数组。参数$number_of_rows 表示指定的记录行，如果没有给出，则一直到 EOF 才停止。

（4）MoveNext()

函数作用：将 ADORecordSet（结果集）的指针下移一位。如果成功，则返回 TRUE，否则返回 FALSE。

（5）Move($to)

函数作用：将 ADORecordSet（结果集）的指针移动到指定位置。如果$to 等于 0，则指针指向结果集的第一条数据；如果$to 的值大于结果集，则指针指向最后一条数据。注意，这里变量$to 只能是绝对定位。

（6）MoveFirst()

函数作用：将指针移动到第一条数据，等同于 Move(0)。

（7）MoveLast()

函数作用：将指针移动到最后一条数据，等同于 Move(RecordCount() - 1)

（8）FetchRow()

函数作用：返回当前指针指向的记录的数组，如果是 EOF，回传 FALSE。FetchRow()不能和 MoveNext()混用。

（9）FetchField($column_number)

函数作用：返回一个对象，包含字段$column_number 的名称、空间长度等相关信息。

参数$column_number 是要查看的字段名称。

（10）FetchNextObject([$toupper=true])

函数作用：返回当前指针所指向的记录的对象形式，并且指针自动下移一行。

参数$toupper 如果等于 TRUE，则字段名为大写形式。

（11）FieldCount()

函数作用：返回结果集里的字段数。

（12）RecordCount( )

函数作用：返回结果集里的记录数。

（13）MetaType($nativeDBType [,$field_max_length],[$fieldobj])

函数作用：标准化不同数据库系统下的数据类型的表示法。处理不同的数据库系统有一个问题，即每一个数据库系统对于相同的数据类型会有不同的表示法，比如支持超长文本的类型，有的表示为 text，有的表示为 long character，还有的表示为 clob 等。MetaType()函数可以将这几种表示方法统一起来，如用大写字母 D 来表示所有的日期类型，用大写字母 T 表示所有的时间类型。

参数说明：

$nativeDBType：数据表中的数据类型。

$fields_max_length：字段最大长度。

$fieldsobj：字段对象。

MetaType()支持的标准化数据类型有：

☑ C:　支持 char、varchar 等字符类型。
☑ X:　支持 text、long character 等长类型。
☑ B:　blob 或 binary image。
☑ D:　日期 date。
☑ T:　时间 timestamp。
☑ L:　逻辑类型和位类型。
☑ N:　包含编号、整型、浮点型等数字类型。
☑ R:　包含序列、自动增加整数等序列类型。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，载入 adodb.ini.php 文件。实现与 MySQL 数据库服务器中 db_database13 数据库的连接，并设置数据库的编码格式为 utf-8。

（2）创建 index.php 文件，包含数据库连接文件，定义 SQL 查询语句，通过 execute()函数执行查询操作，通过 FieldCount ()函数统计字段数量，应用 for 语句和 FetchField()函数循环输出各个字段的值，并应用 metaType()函数对字段中的数据类型进行判断，同时应用 DBDate()函数对时间进行格式化，其关键代码如下：

```
<?php
include_once 'conn/conn.php';                                    //载入数据库连接文件
$count = 1;
$sqlstr = 'select * from tb_object where id=' . $count;          //SQL 查询语句
$rst = $conn->execute ( $sqlstr ) or die ( 'error:   ' . $conn->errorMsg () );   //执行查询语句
if (false != $rst) {                                             //如果有查询结果
        for($i = 0; $i < $rst->FieldCount (); $i ++) {           //循环输出各个字段
                $fields = $rst->FetchField ( $i );               //生成字段信息对象
                $type = $fields->type;
?>
<td bgcolor="#FFFFFF">
<?php
//从对象中获取字段的类型信息
if ($rst->metaType ( $type, - 1, $fields ) == "T")              //如果标准类型为 T
        echo $conn->DBDate ( $rst->fields [$i] );               //使用 DBDate()函数格式化时间
else
        echo $rst->fields [$i];                                  //如果是其他类型，直接输出
?>
</td>
<?php
}
}
?>
```

秘笈心法

心法领悟 543：操作结果集方法的运用。

在本实例中，只是使用 FieldCount ()函数和 FetchField()函数对数据表中的字段进行操作。在实例 544 中，将应用 EOF、fields 和 movenext()函数以及 while 语句完成数据表中数据的循环输出，其实现的是结果集中行的操作。

实例 544 ADODB 控制结果集的存取方法

高级

光盘位置：光盘\MR\13\544

趣味指数：★★★★☆

实例说明

在实例 533 中介绍了很多操作结果集的方法，本实例继续应用上一实例中介绍的操作结果集方法，对数据表中的行进行操作，循环输出数据表中的数据。同时应用一个新的方法，实现对结果集存取方式的控制，其运行结果如图 13.7 所示。



图 13.7　ADODB 类库中对数据库中数据输出的控制

关键技术

在通过对 SQL 语句的操作之后，多数都会返回一个结果集（ADORecordSet 对象）。通过对 ADORecordSet 对象的控制，可以对结果集进行各项操作，这里介绍的是如何控制结果集的存取方法。

1. $ADODB_COUNTRECS

当该变量为 TRUE 时，RecordCount()函数会在数据库驱动不支持 select 指令返回的记录总数时进行自动模拟，返回记录数，默认值为 TRUE。每次进行查询操作时都会自动检查该变量的值。此功能十分消耗内存，不建议使用。

2. $ADODB_CACHE_DIR

该变量设置缓存目录，一般在使用缓存函数时用到，如 CacheExecute()函数。

3. $ADODB_FETCH_MODE

该变量决定结果集以哪种方式进行存取，有 4 个值：

（1）define('ADODB_FETCH_DEFAULT',0);

（2）define('ADODB_FETCH_NUM',1)

结果集以字段序号为索引进行存取，如$rst -> fields[0]、$rst -> fields[1]。

（3）define('ADODB_FETCH_ASSOC',2)

结果集以字段名称为索引进行存取，如$rst -> fields[bigclass]、$rst -> fields[id]。

（4）define('ADODB_FETCH_BOTH',3)

结果集同时支持（2）、（3）两种方式。如果未设置该变量，默认值为（1），不同类型的数据库驱动默认值也不相同。这里不建议使用变量（4），因为有很多的驱动并不支持。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，载入 adodb.ini.php 文件。实现与 MySQL 数据库服务器中 db_database13 数据库的连接，应用$ADODB_FETCH_MODE 设置结果集以字段名称为索引进行存取，并设置数据库的编码格式为 utf-8，其代码如下：

```
<?php
include_once ('../adodb5/adodb.inc.php');                    //载入 adodb.ini.php 文件
$conn = ADONewConnection('mysql');                          //连接 MySQL 数据库
$conn -> PConnect('localhost','root','111','db_database13'); //连接 db_database13 数据库
$ADODB_FETCH_MODE = ADODB_FETCH_ASSOC;                      //设置结果集以字段名称为索引进行存取
$conn -> execute('set names utf8');                         //设置编码格式

?>
```

（2）创建 index.php 文件，完成数据的循环输出。首先，包含数据库连接文件。然后，定义 SQL 查询语句，通过 execute()函数执行查询操作。最后，通过 while 语句循环输出查询结果，其关键代码如下：

```
<?php
 include_once 'conn/conn.php';                              //载入数据库连接文件
 $sqlstr = 'select * from tb_user where id limit 5';        //SQL 查询语句
 $rst = $conn->execute ( $sqlstr ) or die ( 'error:   ' . $conn->errorMsg () ); //执行查询语句
 while ( ! $rst->EOF ) {                                    // wihle 语句循环输出结果
?>
<tr>
        <td width="126" height="30" align="center"><?php echo $rst->fields ['user']; ?></td>
        <td width="117" align="center"><?php echo $rst->fields ['dates']; ?></td>
</tr>
<?php
        $rst->movenext ();                                  //指针下移
 }
 $rst->close ();                                            //关闭连接
 $conn->close ();
 ?>
```

秘笈心法

心法领悟 544：结果集存取方式的选择。

在关键技术中已经对 ADODB 类库中结果集的存取方式进行了详细介绍，至于如何选择，需要根据实际情况具体分析，也可以不设置结果集的存取方式。建议读者以字段名称为索引进行数据存取，这样在获取数据值时更准确一些，不会因为数据表中字段的添加或者删除而影响到获取的值。

实例 545	一次连接两个数据库 光盘位置：光盘\MR\13\545	高级 趣味指数：★★★★☆

实例说明

在实际的程序开发中，如果要实现数据在不同数据库服务器之间的转换，最理想的方法就是编写一个数据转换的程序。这就是本实例中要向大家展示的内容，即同时连接两个不同的数据库服务器，完成数据由 MySQL 数据库到 Access 数据库的转换。运行本实例，单击图 13.8 中的“导出”按钮，将 MySQL 数据库中的数据导入到 Access 数据库中，单击“查看 ACCESS”超链接将看到如图 13.9 所示的存储于 Access 数据库中的数据。

图 13.8　MySQL 数据库中的数据

图 13.9　Access 数据库中的数据

关键技术

一次连接两个数据库，即同时执行两次 ADONewConnection()函数和 PConnect()函数，分别指定不同的 MySQL 数据库服务器和 Access 数据库服务器。

在连接 MySQL 服务器时，设置的用户名是 root，密码是 111，指定的数据库是 db_database13，设置编码格式为 utf-8。

在连接 Access 服务器时，指定的数据库是 db_database13.mdb，注意这里必须设置 Access 数据库文件在服务器中的绝对路径。

同时连接两个数据库的代码如下：

```
<?php
include_once('../adodb5/adodb.inc.php');                    //载入（include）adodb.inc.php 文件
$conn = ADONewConnection('mysql');                           //建立连接
$conn -> PConnect('localhost','root','111','db_database13'); //连接数据库
$conn -> execute('set names utf8');                          //设置编码格式
$con = ADONewConnection("access");                           //连接 Access 数据库
$con-> PConnect("Driver={Microsoft Access Driver (*.mdb)};Dbq=F:\\AppServ\\www\\mr\\13\\008\\conn\\data\\db_database13.mdb");
$con-> execute('set names gb2312');                          //设置编码格式
?>
```

将从 MySQL 数据库中读取的数据添加到 Access 数据库中，应用的是 ADODB 类库中的 GetInsertSQL()函数，该函数用于向数据库中添加新的记录，其语法如下：

```
GetInsertSQL(&$rs, $arrFields)
```

参数$rs 设置要添加记录的结果集；参数$arrFields 设置新记录的字段值及字段名称。

设计过程

（1）创建 conn 文件，编写 conn.php 文件，完成与两个数据库服务器的连接，其代码请参考关键技术中的内容。

（2）创建 index.php 文件，将从 MySQL 数据库中读取出的数据通过 GetInsertSQL()函数添加到 Access 数据库中，并且在本页中输出 MySQL 数据库中的数据，同时创建导出和查看 Aceess 数据库中数据的超链接，其关键代码如下：

```
<?php
include ("conn/conn.php");
$sql = "select * from tb_user where id limit 5 ";                      //创建 SQL 语句
$rst = $conn->execute ( $sql ) or die ( 'error:   ' . $conn->errorMsg () );  //执行查询语句
$array = array ();
while ( ! $rst->EOF ) {
        $array ['id'] = $rst->fields ['id']; # process $arr
        $array ['user'] = iconv ( "utf-8", "gb2312", $rst->fields ['user'] ); # process $arr
        $array ['pass'] = $rst->fields ['pass']; # process $arr
```



关键技术

ADODB 中的 SelectLimit()函数模拟 limit 关键字，控制 select 查询语句的执行。但要注意，SelectLimit()函数的参数位置与 MySQL 数据库中 limit 关键字是不同的。SelectLimit()函数的语法如下：

```
SelectLimit($sql[,$numrows=-1][,$offset=-1][,$inputarr=false])
```

参数说明：

$sql：要执行的 select 查询语句。

$numrows：要查询的记录数。如果该值为-1，则查询到最后一条记录。

$offset：表示从第几条记录开始查询。

$inputarr：结合变量。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库服务器中 db_database13 数据库的连接。

（2）创建 index.php 文件，包含数据库连接文件，定义 SQL 查询语句，应用 SelectLimit()函数查询出班级前 3 名学生的成绩，并且循环输出查询结果，其关键代码如下：

```
<?php
include_once ('conn/conn.php');                                   //载入 adodb.ini.php 文件
$sql="select * from tb_student where id order by grade desc";     //定义查询语句
$numrows = 3;                                                     //设置查询的记录数
$offset = 0;                                                      //设置起始位置
$rst = $conn -> SelectLimit($sql,$numrows,$offset) or die('execute error：'.$conn -> ErrorMsg());  //执行查询
while(!$rst -> EOF){
?>
<tr>
    <td height="25" align="center"><?php echo $rst -> fields['studentname'];?></td>
    <td align="center"><?php echo $rst -> fields['classname']; ?></td>
    <td align="center"><?php echo $rst -> fields['grade']; ?></td>
</tr>
<?php
$rst -> movenext();                                               //指针下移
}
$rst -> close();                                                  //关闭连接
$conn -> close();
?>
```

秘笈心法

心法领悟 546：明确 SelectLimit()函数与 MySQL 数据库中 limit 关键字的区别。

在 SelectLimit()函数中，$numrows 指定的是要查询的记录数，而$offset 指定从第几条记录开始查询，而在 MySQL 数据库中，limit 关键字指定查询的开始位置，而 offset 指定查询的记录数。

实例 547 ADODB 生成 HTML 表格

光盘位置：光盘\MR\13\547

高级

趣味指数：★★★★☆

实例说明

在 ADODB 类库中，有一种方法可以返回表格式的结果集，即将查询结果直接以表格的形式进行输出。本实例就应用这种方法循环输出编程词典的数据，其运行结果如图 13.11 所示。

序号	名称	类别	时间
1	VC编程词典	标准版	Thu 29, Jul 2010
2	VB编程词典	标准版	Thu 29, Jul 2010
3	JAVA编程词典	标准版	Thu 29, Jul 2010

图 13.11　ADODB 生成 HTML 表格

关键技术

本实例应用 rs2html()函数，返回一个 HTML 表格格式的结果集，其语法如下：

```
function rs2html($adorecordset,[$tableheader_attributes], [$col_titles])
```

参数说明：

$adorecordset：要返回的结果集。

$tableheader_attributes：对表格的参数及属性的设置。

$col_titles：对字段的重新命名。

要运用本函数，必须载入 adodb 类库中的 tohtml.inc.php 文件。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

（2）创建 index.php 文件，包含数据库连接文件，定义 SQL 查询语句，应用 rs2html()函数将查询结果以表格的形式输出，其关键代码如下：

```
<?php
include_once 'conn/conn.php';                                      //载入数据库链接文件
include_once '../adodb5/tohtml.inc.php';                           //载入 tohtml.inc.php 文件
$rst = $conn -> execute('select * from tb_bccd');                  //返回查询结果集
rs2html($rst,' width="775" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#CCCCCC"',array('序号','名称','类别','时间
'));
?>
```

秘笈心法

心法领悟 547：rs2html()函数。

这是一个独立的函数（rs2heml = recordset to html），相当于 PHP 中的 odbc_result_all()函数。它输出一个完整的 ADORecordSet，参数$adorecordset 如同一个 HTML 表格；参数$tableheader_attributes 允许控制表格中的参数，例如 cellpadding、cellspacing 或者 border 等属性。最后，通过$col_titles 参数，设置数据表中字段的名称。注意，该函数是一种快速除错机制，并不是一个理想的结果集输出方法。

实例 548　ADODB 生成下拉列表框

光盘位置：光盘\MR\13\548

高级

趣味指数：★★★★

实例说明

在 ADODB 中，有一种方法可以将结果集中的数据直接以下拉列表框的形式返回，这种方法非常适合论坛中帖子标题或者类别的输出，也可以在登录用户的权限选择中使用。在本实例中，应用这种方法完成用户登录时编程词典类型的选择，其运行结果如图 13.12 所示。

图 13.12　ADODB 生成下拉列表框

关键技术

ADODB 中生成下拉列表框应用的是 GetMenu()函数。

GetMenu()函数，返回一个 HTML 下拉列表框（select）。结果集的第一列（fields[0]），被显示到下拉列表当中。第二列（fields[1]）被设置为<option>标签的 value 属性，其语法如下：

```
GetMenu($name, [$default_str=''], [$blank1stItem=true], [$multiple_select=false], [$size=0], [$moreAttr=''])
```

参数说明：

$name：下拉列表框的名称。

$default_str：如果$default_str 的值等于其中一个 fields[0]，那么这个 fields[0]就是该下拉列表的默认选项。相当于为这个 fileds[0]添加了属性<selected>。

$blank：该参数默认值为 true。在生成下拉列表框时，第一行选项默认为空，即<option></option>。如果为 false，则没有默认的空选项。

$multiple_select：是否支持多选，默认值为 false，即不允许。

$size：选取列表框的大小，默认为 0，生成一个下拉列表。如果$size 的值大于 0，则生成一个列表框。

$moveAttr：该参数可以增加列表框的属性，如添加一个 javascript 脚本等。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

（2）编写 index.php 文件，创建 form 表单，设置 action 属性值为 index_ok.php。包含数据库连接文件，定义 SQL 查询语句，应用 GetMenu2()函数将查询结果以下拉列表框的形式输出，其关键代码如下：

```
<?php
include_once 'conn/conn.php';                                              //载入数据库连接文件
$sqlstr = 'select name,id from tb_bccd';                                   //SQL 查询语句
$rst = $conn -> execute($sqlstr) or die('execute error:   '.$conn -> errorMsg());   //执行查询语句，返回结果集
?>
<!--   调用 getMenu2()函数   -->
<?php echo $rst -> getMenu2('type',$_POST['name'],false); ?>
```

（3）创建 index_ok.php 文件，获取表单中提交的数据，完成用户的登录操作。

秘笈心法

心法领悟 548：生成下拉列表框的另一个函数——GetMenu2()。

在本实例中，我们运用的并不是关键技术中讲解的 GetMenu()函数，而是 GetMenu2()函数。GetMenu2()函数的作用与 GetMenu()函数相同，它们之间的区别是：GetMenu()中的参数$default_str 和 fields[0]比较，如果相等，那么 fields[0]就是这个下拉列表的默认选项；而 getMenu2()中的参数$default_str 则是和 fields[1]进行比较，其他不变。

实例 549	事务处理机制保证数据库操作的完整性 光盘位置：光盘\MR\13\549	高级 趣味指数：★★★☆

实例说明

事务处理机制在程序开发过程中有着非常重要的作用，它可以使整个系统更加安全，例如在银行处理转账业务时，如果 A 账户中的金额刚被发出，而 B 账户还没来得及接受就发生停电事故，这会给银行和个人带来很大的经济损失。采用事务处理机制，一旦在转账过程中发生意外，则程序将回滚，不做任何处理。在本实例中，沿用实例 545 的内容，实现数据在不同数据库间的转换功能，在将 MySQL 数据库中的数据转换到 Access 数据

库中后，删除 MySQL 数据库中的原始数据，并应用事务处理机制保证数据转换操作安全、顺利地进行。运行本实例，如图 13.13 所示，单击“数据转移”超链接，将 MySQL 数据库中的数据转移到 Access 数据库中，并且删除 MySQL 数据库中的原始数据。单击“查看 ACCESS”超链接，将查看到 Access 数据库中存储的数据，如图 13.14 所示。

图 13.13　数据转移

图 13.14　查看 Access 数据库中的数据

关键技术

在 ADODB 中，事务处理应用的是如下 3 个函数：

（1）BeginTrans()函数，开启事务处理。该函数和下面的两个函数都是应用在 PHP 的事务处理中。

（2）CommitTrans()函数，如果成功完成数据库操作，则执行该函数。

（3）RollbackTrans()函数，结束一次操作，恢复操作前的所有改变。成功返回 TRUE，如果服务器不支持返回 FALSE。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

（2）编写 index.php 文件，包含数据库连接文件，定义 SQL 语句，循环输出 MySQL 数据库中的数据，通过 BeginTrans()函数开启事务处理。然后，根据超链接传递的参数进行判断，通过 GetInsertSQL()函数执行数据的转移操作，转移成功后通过 delete 语句删除原始数据库中的数据。最后，根据添加语句和删除语句的执行情况，判断是执行 CommitTrans ()提交操作，还是执行 RollbackTrans ()回滚操作，其关键代码如下：

```
<?php
include ("conn/conn.php");
$sql = "select * from tb_user where id order by id limit 5 ";                    //创建 SQL 语句
$rst = $conn->execute ( $sql ) or die ( 'error:  ' . $conn->errorMsg () );       //执行查询语句
$array = array ();
$con->BeginTrans ();                                                             //开始事务处理
$conn->BeginTrans ();                                                            //开始事务处理
while ( ! $rst->EOF ) {
        if ($_GET ['insert'] == "数据转移") {
                $array ['id'] = $rst->fields ['id']; # process $arr
                $array ['user'] = iconv ( "utf-8", "gb2312", $rst->fields ['user'] ); # process $arr
                $array ['pass'] = $rst->fields ['pass']; # process $arr
                $array ['dates'] = $rst->fields ['dates']; # process $arr
                $sqls = "select * from tb_user where id=0 ";                     //创建 SQL 语句
                $ret = $con->execute ( $sqls ) or die ( 'error:  ' . $con->errorMsg () );    //执行查询语句
                $insert = $con->getInsertSQL ( $ret, $array ) or die ( 'update error: ' . $con->errorMsg () );   //添加新数据
                $result=$con->execute ( $insert );
                $sql = 'delete from tb_user where id =' . $rst->fields ['id'];   //SQL 删除语句
                $results = $conn->execute ( $sql ) or die ( 'execute error:  ' . $conn->ErrorMsg () );   //执行删除语句
        }
?>
                <tr>
```







```
<?php
error_reporting(E_ALL);                                        //显示所有的错误信息
include_once ('../adodb5/adodb-errorhandler.inc.php');
include_once ('../adodb5/adodb.inc.php');                      //载入 adodb.ini.php 文件
$conn = ADONewConnection('mysql');                             //连接 MySQL 数据库
$conn -> PConnect('localhost','root','111','db_database13');   //连接 db_database13 数据库
//$ADODB_FETCH_MODE = ADODB_FETCH_ASSOC;                       //设置结果集以字段名称为索引
$conn -> execute('set names utf8');                            //设置编码格式
?>
```

（2）编写 index.php 文件。首先，包含数据库连接文件。然后，定义 SQL 语句，通过 execute()函数执行 SQL 语句。最后，循环输出 MySQL 数据库中的数据，其关键代码如下：

```
<?php
        include ("conn/conn.php");
        $sql = "select * from tb_use where id limit 5 ";       //创建 SQL 语句
        $ret = $conn -> execute($sql);                          //执行查询语句
        while ( ! $ret->EOF ) {                                 //循环输出
?>
<tr>
        <td height="20"><?php echo $ret->fields[0];     ?></td>
        <td height="20"><?php echo $ret->fields[1];     ?></td>
        <td><?php echo $ret->fields[3];     ?></td>
</tr>
<?php
        $ret->movenext ();                                      //指针下移
}
$ret->close ();                                                 //关闭连接
$conn->close ();
?>
```

秘笈心法

心法领悟 552：通过 error_reporting()函数控制返回的错误信息。

在本实例中，设置 error_reporting()函数的值为 E_ALL 返回所有的错误信息。如果将值设置为 0，将不会有任何错误信息输出。如果想更多地了解 error_reporting()函数，请参考 PHP 函数手册。

实例 553	将错误信息存储于日志文件中 光盘位置：光盘\MR\13\553	高级 趣味指数：★★★★☆

实例说明

在实例 552 中，只是应用 ADODB 类库中提供的自定义错误处理机制返回所有的错误信息，在本实例中将介绍另外一种错误处理机制，不但可以返回错误信息，而且可以将错误信息存储到指定的文件中。在本实例中，将返回的错误信息存储于根目录下 error 文件夹中，error.log 中的错误信息如图 13.18 所示。

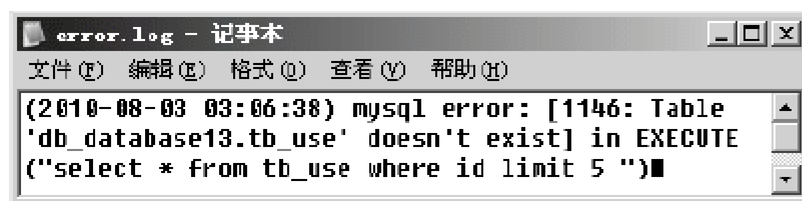


图 13.18　通过日志文件存储错误信息

关键技术

同样，应用 ADODB 中提供的自定义错误处理机制时，必须在程序连接数据库服务器之前载入 adodb-errorhandler.inc.php 文件。

将错误信息存储于指定的文件中，应用的是两个选择性常量：ADODB_ERROR_LOG_TYPE 和 ADODB_ERROR_LOG_DEST。

将 ADODB_ERROR_LOG_TYPE 的值设置为 3，即将错误信息存储于常量 ADODB_ERROR_LOG_DEST 指定的文件中。这里指定 ADODB_ERROR_LOG_DEST 的值为 error/error.log，即将错误信息存储于实例根目录

下 error 文件夹的 error.log 文件中。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件。首先，定义两个常量，设置程序中错误信息的存储位置。然后，载入 adodb-errorhandler.inc.php 文件。最后，完成与 MySQL 数据库的连接，其代码如下：

```
<?php
define('ADODB_ERROR_LOG_TYPE',3);                              //设置类型
define('ADODB_ERROR_LOG_DEST','error/error.log');              //设置存储路径
include_once ('../adodb5/adodb-errorhandler.inc.php');         //载入自定义错误处理文件
include_once ('../adodb5/adodb.inc.php');                      //载入 adodb.ini.php 文件
$conn = ADONewConnection('mysql');                             //连接 MySQL 数据库
$conn -> PConnect('localhost','root','111','db_database13');   //连接 db_database13 数据库
$conn -> execute('set names utf8');                            //设置编码格式
?>
```

（2）编写 index.php 文件。首先，包含数据库连接文件。然后，定义 SQL 语句，通过 execute()函数执行 SQL 语句。最后，循环输出 MySQL 数据库中的数据，其关键代码如下：

```
<?php
    include ("conn/conn.php");
    $sql = "select * from tb_use where id limit 5 ";           //创建 SQL 语句
    $ret = $conn -> execute($sql);                             //执行查询语句
    while ( ! $ret->EOF ) {                                    //循环输出
?>
<tr>
    <td height="20"><?php echo $ret->fields[0];    ?></td>
    <td height="20"><?php echo $ret->fields[1];    ?></td>
    <td><?php echo $ret->fields[3];    ?></td>
</tr>
<?php
    $ret->movenext ();                                         //指针下移
}
$ret->close ();                                                //关闭连接
$conn->close ();
?>
```

秘笈心法

心法领悟 553：ADODB_ERROR_LOG_TYPE 常量设置错误记录的类型。

用 ADODB_ERROR_LOG_TYPE 常量设置错误记录的类型，可选值如下：

☑ 0，默认值。根据 php.ini 文件中的 error_log 配置，错误被发送到服务器日志系统或文件。

☑ 1，错误被发送到 destination 参数中的地址。只有该类型使用 headers 参数。

☑ 2，通过 PHP debugging 连接来发送错误。该选项只在 PHP 3 中可用。

☑ 3，错误被发送到文件目标字符串。

13.4　ADODB 操作数据库

实例 554	应用 ADODB 访问 Access 数据库 光盘位置：光盘\MR\13\554	高级 趣味指数：★★★☆

实例说明

虽然 PHP 是构建 Web 系统强有力的工具，但是由于 PHP 存取函数一直没有标准化，不同数据库间的函数名称、参数差异很大，在更换数据库时，会带来大量的代码修复工作。这时，就需要一组函数库来隐藏不同数

据库函数间的差异，让开发者可以很简单地去切换数据库，这就是 ADODB 类库。

本实例中，应用 ADODB 访问 Access 数据库，并通过分页函数，实现数据库中数据的分页输出，其运行结果如图 13.19 所示。



图 13.19　ADODB 分页输出 Access 数据库中的数据

关键技术

应用 ADODB 操作 Access 数据库最简单的方法就是应用在实例 538 和实例 539 中定义的 ADODB 连接和操作数据库类。

本实例的目的是让读者更深刻地理解 ADODB 操作 Access 数据库的方法和步骤，如下所示：

（1）载入 adodb.inc.php 文件，这是应用 ADODB 类库操作任何数据库的前提。

（2）连接 Access 数据库服务器，依然应用 ADONewConnection()函数，只是其参数值是 Access。

（3）应用 PConnect()函数连接指定的数据库文件。这是 Access 数据库与其他数据库不同的地方，数据库文件的位置必须是一个绝对路径。

（4）设置数据库的编码格式。

ADODB 连接 Access 数据库的关键代码如下：

```
<?php
include('../adodb5/adodb.inc.php');                                    //载入（include）adodb.inc.php 文件
$conn = ADONewConnection('access');                                    //连接 Access 数据库
$conn-> PConnect("Driver={Microsoft Access Driver (*.mdb)};Dbq=F:\\AppServ\\www\\mr\\13\\017\\data\\db_database13.mdb");
$conn -> execute('set names gb2312');                                  //设置编码格式
?>
```

（5）根据连接数据库成功返回的对象，执行查询语句，完成数据的分页输出，有关数据的分页输出方法请读者参考实例 540 和实例 541，这里不再赘述。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件。连接 Access 数据库，其代码请参考关键技术中的内容。

（2）创建 index.php 文件。首先，包含数据库连接文件，载入 tohtml.inc.php 文件。然后，定义 SQL 语句，定义分页变量。最后，应用分页函数完成数据的分页输出，其关键代码如下：

```
<?php
include("conn/conn.php");                                              //载入数据库连接文件
include('../adodb5/tohtml.inc.php');                                   //载入 tohtml.inc.php 文件
$sql = 'select * from tb_bookinfo';                                    //查询语句
$num =10;                                                              //每页显示的记录数
if(isset($_GET['n_page'])){                                            //判断当前页码
        $c_page = $_GET[n_page];                                       //将$n_page 赋给变量$c_apge
}else{
        $c_page = 1;                                                   //初始化变量$c_page
}
$rst = $conn -> PageExecute($sql,$num,$c_page);                        //执行 PageExecute 函数
rs2html($rst,'width=810 bordercolor=#FFFFFF ',array('图书名称','出版社','页码','价格','出版时间','ID'));
?>
<table width="520" height="24" border="0" cellpadding="0" cellspacing="0" bordercolor="#FFFFFF" bgcolor="#999999">
```

```
  <tr>
    <td><font color='#999999'>当前是第<?php echo $rst -> absolutePage(); ?>页/一共是<?php echo $rst -> LastPageNo(); ?>页</font></td>
    <td align="center" bgcolor="#FFFFFF">
<?php
if(false !== $rst){
        if(!$rst -> AtfirstPage()){                                   //如果当前页不是首页
?>
<!--   输出向上翻页超链接   -->
            <a href ="<?php echo '?n_pge=1' ?>"> 首页 </a>
            <a href ="<?php echo '?n_page='.($rst -> absolutePage() - 1); ?>"> 上一页 </a>
<!--   ---------------------------   -->
<?php
        }
        if(!$rst -> AtlastPage()){                                    //如果当前页不是尾页
?>
<!--   输出向下翻页超链接   -->
            <a href = "<?php echo '?n_page='.($rst -> absolutePage() + 1); ?>"> 下一页 </a>
            <a href ="<?php echo '?n_page='.($rst -> LastPageNo());?>"> 尾页 </a>
<!--   ----------------------------   -->
<?php
        }
}
?>
```

秘笈心法

心法领悟 554：ADODB 操作 Access 数据库必须注意的问题。

在连接数据库时，指定的 Access 数据库文件的存储位置必须是绝对路径。另外，从 Access 数据库中读取的数据要使用 gb2312 编码，不可以使用 UTF 编码格式。如果页面中使用的是 utf-8 编码，那么可以在输出数据时应用 iconv()函数对字符串的编码格式进行转换。

实例 555	应用 ADODB 访问 MySQL 数据库 光盘位置：光盘\MR\13\555	高级 趣味指数：★★★★☆

实例说明

本实例介绍应用 ADODB 访问 MySQL 数据库，并且通过 while 语句循环输出数据库中的数据，运行结果如图 13.20 所示。

图 13.20　ADODB 访问 MySQL 数据库

关键技术

在本实例中，应用 ADONewConnection()函数连接数据库服务器，应用 PConnect()函数连接 db_database13 数据库，设置用户名为 root，密码为 111，设置数据库编码格式为 utf-8。

通过 execute()函数执行 SQL 语句，在 while 循环语句中，应用 EOF 判断指针位置，应用 fields()函数获取数据表中对应字段的值，应用 movenext()函数控制指针的移动。

本实例中应用的函数请读者参考实例 543 和实例 544 关键技术中的内容，这里不再赘述。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 MySQL 数据库的连接。

（2）创建 index.php 文件。首先，包含数据库连接文件。然后，定义 SQL 语句，通过 execute()函数执行 SQL 语句。最后，通过 while 循环输出 MySQL 数据库中的数据，其关键代码如下：

```
<?php
include_once ('conn/conn.php');                              //载入（include）adodb.inc.php 文件
$sqlstr = 'select * from tb_bccd';                           //创建 SQL 语句
$rst = $conn -> execute($sqlstr) or die('connect error');    //执行 SQL 语句，查询出 id 等于 2 的数据
while(!$rst -> EOF){                                         // wihle 语句循环输出结果
?>
  <tr>
    <td   height="20" bgcolor="#FFFFFF"><?php echo $rst -> fields['id'];?></td>
    <td bgcolor="#FFFFFF"><?php echo $rst -> fields['name'];?></td>
    <td bgcolor="#FFFFFF"><?php echo $rst -> fields['type'];?></td>
    <td bgcolor="#FFFFFF"><?php echo $rst -> fields['dates'];?></td>
  </tr>
<?php
      $rst -> movenext();                                    //指针下移
}
$rst -> close();                                             //关闭连接
$conn -> close();                                            //关闭数据库
?>
```

秘笈心法

心法领悟 555：选择适合的 ADODB 操作数据库方法。

在实际的 Web 程序开发过程中，建议读者应用实例 538 和实例 539 中封装的 ADODB 连接和操作数据库类来实现对数据库的操作。因为这种方法更适合对数据库的管理、维护和更新，一旦出现更换数据库服务器的问题也容易解决。本实例中介绍的单一的数据库连接、操作方法，存在一定的局限性，更适合于读者在学习阶段使用，以了解 ADODB 操作数据库的函数和步骤。

实例 556　应用 ADODB 访问 SQL Server 数据库　高级

光盘位置：光盘\MR\13\556

趣味指数：★★★☆

实例说明

本实例介绍应用 ADODB 访问 SQL Server 数据库，并且通过 while 循环语句输出数据库中的数据，其中还应用$ADODB_FETCH_MODE 变量控制结果集的存储方式，运行结果如图 13.21 所示。

商品名称	添加时间	数量	价格
显示器	2009-05-23 00:00:00.000	25	1200.0000
主板	2009-05-23 00:00:00.000	12	600.0000
显卡	2009-05-23 00:00:00.000	5	450.0000

图 13.21　ADODB 访问 SQL Server 数据库

关键技术

通过 ADODB 访问 SQL Server 数据库的操作步骤与实例 555 应

用 ADODB 访问 MySQl 数据库是相同的，这里就不再赘述，唯一的区别是在访问 SQL Server 数据库时，应用 $ADODB_FETCH_MODE 变量对结果集的存取方式进行控制。

$ADODB_FETCH_MODE 变量决定结果集以哪种方式进行存取，有关该变量的详细讲解请参考实例 544。

在通过 ADODB 连接 SQL Server 数据库时还有一点需要注意，即在通过 PConnect()函数连接数据库文件时，对 Server 服务器的设置。如果 SQL Server 数据库存储于本机中，那么 Server 的参数值可以使用 localhost。但是，如果 SQL Server 数据库存储在其他机器中，那么 Server 的参数值必须是 SQL Server 的服务器名称。

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与 SQL Server 数据库的连接，并且设置以字段名称为结果集索引。

（2）编写 index.php 文件。首先，包含数据库连接文件。然后，定义 SQL 语句，通过 execute()函数执行 SQL 语句。最后，循环输出 SQL Server 数据库中的数据，其关键代码如下：

```
<?php
include_once("conn/conn.php");
$sqlstr = "select * from tb_goods where addtime='2009-5-23' ";              //创建 SQL 语句
$rst = $conn -> execute($sqlstr) or die('connect error');                  //执行 SQL
while(!$rst -> EOF){                                                        //如果没有错误，则配合 wihle 循环输出结果
?>
    <tr>
      <td align="left" bgcolor="#FFFFFF"> <?php echo iconv('gb2312',"utf-8",$rst -> fields['name']);?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $rst -> fields['addtime'];?></td>
      <td align="left" bgcolor="#FFFFFF"> <?php echo $rst -> fields['counts'];?></td>
      <td align="center" bgcolor="#FFFFFF"><?php echo $rst -> fields['price'];?></td>
    </tr>
<?php
      $rst -> movenext();                                                   //指针下移
}
$rst -> close();                                                            //关闭连接
$conn -> close();
?>
```

秘笈心法

心法领悟 556：了解 ADODB 操作数据库的方法和步骤。

在本节的 3 个实例中，分别介绍了如何通过 ADODB 操作 Access、MySQL 和 SQL Server 数据库，其目的是让读者对 ADODB 中操作数据库的方法和步骤有一个系统的了解，从中读者可以体会到，无论是操作哪种数据库，所使用的方法和操作步骤基本都是相同的，这就是 ADODB 类库的强大之处，这也就是笔者为什么推崇在实际的程序开发过程中使用实例 538 和实例 539 中封装的类来连接和操作数据库的原因。

第14章

Smarty 模板

⏭ 环境搭建

⏭ 输出数据

⏭ 综合应用

14.1 环 境 搭 建

实例 557 **Smarty 开发环境搭建** 高级

光盘位置：光盘\MR\14\557 趣味指数：★★★★☆

实例说明

PHP 没有内置 Smarty 模板类，需要单独下载和配置，并且 Smarty 要求服务器上的 PHP 版本最低为 4.0.6。用户可以去网站 http://smarty.net/download.php 下载最新的 Smarty 压缩包。本书使用的版本是 Smarty-2.6.23，下载地址如图 14.1 所示。



图 14.1 Smarty 模板下载页面

关键技术

Smarty 模块的安装步骤：

（1）将下载的 Smarty 压缩包解压，解压后的文件夹如图 14.2 所示，其中有一个 libs 文件夹，包含了 4 个 Smarty 类库的核心文件：smarty.class.php、smarty_Compiler.class.php、config_File.class.php 和 debug.tpl，以及 internals 和 plug-ins 两个文件夹，如图 14.3 所示。



图 14.2 Smarty-2.6.23 的压缩包

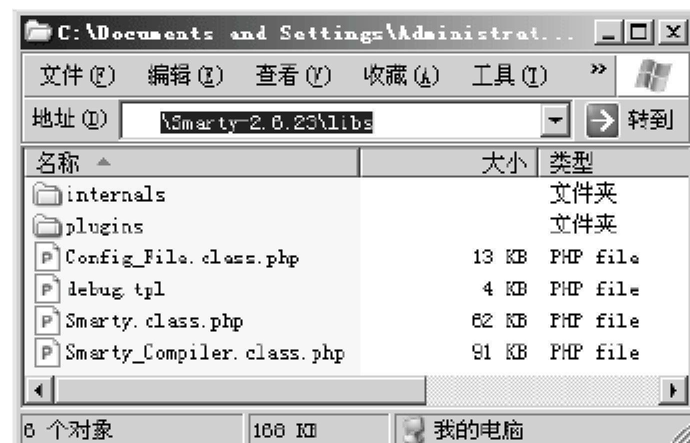


图 14.3 Smarty-2.6.23 的 libs 文件





图 14.6　第一个 Smarty 程序

（3）创建配置文件。如果要应用 Smarty 模板，那么就一定要包含 Smarty 类库和相关信息。配置信息可以存储在一个单独的文件中，在需要时通过 include 语句包含即可。配置文件 config.php 的代码如下：

```
<?php
    /*   定义服务器的绝对路径   */
    define('BASE_PATH',$_SERVER['DOCUMENT_ROOT']);
    /*   定义 Smarty 模板的绝对路径   */
    define('SMARTY_PATH','\mr\14\Smarty\\');
    /*   加载 Smarty 类库文件       */
    require BASE_PATH.SMARTY_PATH.'Smarty.class.php';
    /*   实例化一个 Smarty 对象   */
    $smarty = new Smarty;
    /*   定义各个目录的路径       */
    $smarty->template_dir = './';                        //定义模板文件的存储位置
    $smarty->compile_dir = './Smarty/templates_c/';      //定义编译文件的存储位置
    $smarty->config_dir = './Smarty/configs/';           //定义配置文件的存储位置
    $smarty->cache_dir = './Smarty/cache/';              //定义缓存文件的存储位置
?>
```

上述配置文件的参数说明如下：

BASE_PATH：指定服务器的绝对路径。

SMARTY_PATH：指定 Smarty 模板的绝对路径。

require()：加载 Smarty 类库文件 Smarty.class.php。

$smarty：实例化 Smart 对象。

$smarty->template_dir：定义模板文件存储位置。

$smarty-> compile_dir：定义编译文件存储位置。

$smarty-> config_dir：定义配置文件存储位置。

$smarty-> cache_dir：定义缓存文件存储位置。

设计过程

完成 Smarty 模板的配置后，编写一个实例，体会 Smarty 模板引擎的特性。

（1）在实例 558 文件夹下创建 index.php 文件，编写动态 PHP 代码。首先，设置页面的编码格式。然后，通过 include 语句包含 Smarty 的配置文件。最后，通过 Smarty 类库中的 assign()方法将数据赋给模板变量，并且应用 display 指定模板页，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );         //设置页面编码格式
include("config.php");                                        //包含配置文件
/*   使用 Smarty 赋值方法将一对儿名称/方法发送到模板中   */
$smarty->assign('title','第一个 Smarty 程序');
```

```
$str = file_get_contents('files/content.txt');                    //读取文本文件中的数据
$smarty->assign('content',iconv("gb2312","utf-8",$str));          //将文本文件中的数据存储到模板变量中
/*   显示模板   */
$smarty->display('index.html');
?>
```

☑ assign()方法用于在模板被执行时为模板变量赋值，语法如下：

```
{assign var=" " value=" "}
```

参数 var 是被赋值的变量名，参数 value 是赋给变量的值。

☑ display()方法用于显示模板，需要指定一个合法的模板资源的类型和路径，还可以通过第 2 个可选参数指定一个缓存号，相关的信息可以查看缓存。

```
void display (string template [, string cache_id [, string compile_id]])
```

参数 template 指定一个合法的模板资源的类型和路径；参数 cache_id 为可选参数，指定一个缓存号；参数 compile_id 为可选参数，指定编译号。编译号可以将一个模板编译成不同版本使用，例如针对不同的语言编译模板。编译号的另外一个作用是，如果存在多个$template_dir 模板目录，但只有一个$compile_dir 编译后存档目录，这时可以为每一个$template_dir 模板目录指定一个编译号，以避免相同的模板文件在编译后互相覆盖。相对于在每一次调用 display()的时候都指定编译号，也可以通过设置$compile_id 编译号属性来一次性设定。

（2）在实例 558 文件夹下创建 Smarty 文件夹，并在该文件夹下新建 3 个文件夹：templates_c、configs 和 cache。

（3）在实例 558 文件夹下创建 index.html 文件，作为 Smarty 的模板文件，在该文件中输出模板变量传递的数据，其关键代码如下：

```
<!--输出标题-->
<title>{$title}</title>
<!--输出内容-->
<p>{$content}</p>
```

至此，第一个应用 Smarty 模板开发的实例编写完成，在设计过程中只讲解了 3 个步骤，其实编写一个完整的 Smarty 模板实例需要 5 个步骤：

（1）下载 Smarty 类库文件，并存储到指定位置。

（2）编程 Smarty 模板的配置文件。

（3）创建 templates、templates_c、configs 和 cache 文件夹，设置 Smarty 中各种文件的存储位置。

（4）创建动态 PHP 文件，编写 PHP 代码。

（5）编写 Smarty 模板文件，设计静态页面，并且输出 Smarty 模板变量中传递的数据。

之所以在设计过程中给出 3 个步骤，是因为其他两个步骤已经在关键技术中进行了详细的讲解。

这是一个简单的 Smarty 模板应用实例，读者也许体会不到 Smarty 模板引擎的强大功能及给程序开发带来的便利，但是通过这个实例已经展示出 Smarty 应用的基本原理和操作流程。

秘笈心法

心法领悟 558：Smarty 模板引擎的特性。

Smarty 是一个使用 PHP 编写的应用于 PHP 的模板引擎，它将一个应用程序分成两部分：视图和逻辑控制。简单地讲，就是将 UI（用户界面）和 PHP code（PHP 代码）分离。这样，程序员在修改程序时，不会影响到页面设计，而美工在重新设计或修改页面时，也不会影响程序逻辑。

实例 559　封装 Smarty 模板的配置方法

光盘位置：光盘\MR\14\559

高级　趣味指数：★★★☆

实例说明

在实例 558 中讲解的 Smarty 模板配置方法有一个弊端，即如果存储实例的文件夹发生变化，那么就必须修

改 config.php 配置文件中常量 SMARTY_PATH 的值，否则将找不到 Smarty 类库的存储位置。例如，将 mr 文件夹修改成 mrs 文件夹，那么在运行实例 558 时，将输出如图 14.7 所示的错误信息。

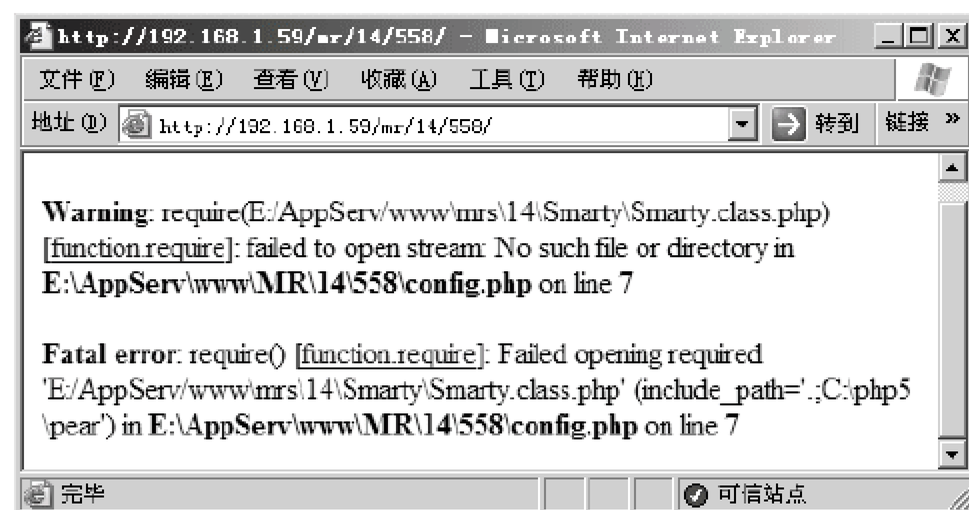


图 14.7　找不到指定的 Smarty 类库文件

本实例向读者介绍另外一种 Smarty 模板的配置方法，使用这种方法，无论怎么改变实例的文件夹，都不会影响实例的运行。

关键技术

这里将 Smarty 配置方法封装到类中，并存储到 system.smarty.inc.php 文件中。首先，包含 Smarty 类文件 Smarty.class.php。然后，定义 SmartyProject 类，继承 Smarty 父类。最后，定义 SmartyProject()方法，设置 Smarty 中模板文件（templates）、编译文件（templates_c）、配置文件（configs）和缓存文件（cache）的存储位置。这就是新的配置文件 system.smarty.inc.php，其代码如下：

```
<?php
require("../Smarty/Smarty.class.php");                    //调用 Smarty 文件
class SmartyProject extends   Smarty{                     //定义类，继承 Smarty 父类
 function SmartyProject(){                                //定义方法，配置 Smarty 模板
        $this->template_dir = "./";                       //指定模板文件存储在根目录下
        $this->compile_dir = "./system/Smarty/templates_c/";  //指定编译文件存储位置
        $this->config_dir = "./system/Smarty/configs/";
        $this->cache_dir = "./system/Smarty/cache/";
    }
}
?>
```

既然已经将 Smarty 的配置方法存储到一个类中，那么就需要对类进行实例化，根据返回的对象名称调用 Smarty 中的方法，类的实例化操作在 system.inc.php 文件中完成，其返回对象名为$smarty，其代码如下：

```
<?php
require("system.smarty.inc.php");                         //调用类文件
$smarty=new SmartyProject();                              //执行类的实例化操作
?>
```

通过此方法配置 Smarty 模板的好处是，无论将程序复制到哪个服务器下执行，都不需要更改服务器或者 Smarty 文件的绝对路径，程序可以直接运行。

技巧： 这种配置方法中调用的仍是存储在章文件夹 14 下的 Smarty 类库，同时在这种方法中将配置文件存储于实例根目录下的 system 文件夹中，将模板文件夹（templates）、编译文件夹（templates_c）、配置文件夹（configs）和缓存文件夹（cache）存储在 system\Smarty 目录下。

设计过程

下面应用新的配置方法对实例 558 进行重新编写，生成实例 559，步骤如下：

（1）在实例的根目录下创建 system 文件夹，存储 Smarty 的配置文件 system.smarty.inc.php 和实例化操作文件 system.inc.php，同时创建 Smarty 文件夹，最后在 Smarty 文件夹下创建 3 个文件夹 templates_c、configs 和 cache。

（2）编写 index.php 文件，重新载入 Smarty 的配置文件，即通过 include 语句包含 system.inc.php。

（3）其他内容与实例 558 相同，这里不再赘述。

为了让读者更好地理解 Smarty 模板的配置方法和文件的存储位置，下面给出本实例的文件夹架构，如图 14.8 所示。

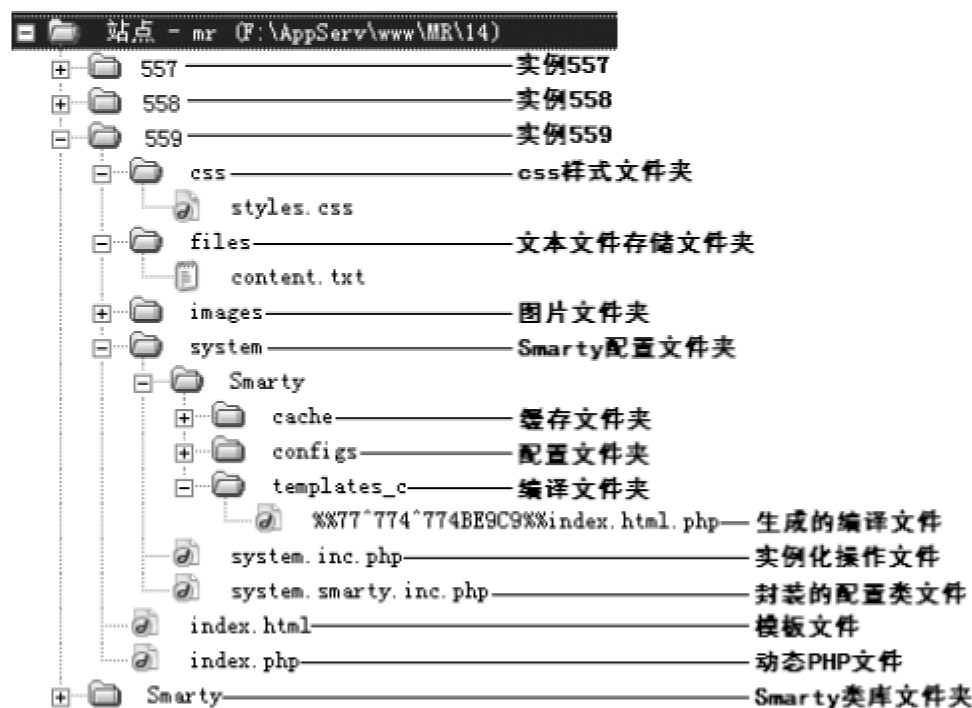


图 14.8　实例 559 的文件夹架构

秘笈心法

心法领悟 559：确立 Smarty 类库文件的存储位置。

在本章中，我们将 Smarty 类库文件存储在章文件夹 14 下，本章中所有的实例调用的都是这个 Smarty 类库，并且从本实例开始，以后的实例都采用本实例中的 Smarty 配置方法，以后不再赘述。

另外，如果要运行本章中的某个实例（实例 558 除外），必须将 Smarty 类库与实例文件夹放置于同级目录下，并且设置 Smarty 类库的名称为 Smarty。

14.2　输 出 数 据

实例 560	通过 if 语句判断当前用户的权限 光盘位置：光盘\MR\14\560	高级 趣味指数：★★★★☆

实例说明

在动态 PHP 文件中，可以通过 SESSION 变量的值判断当前用户是否具有访问权限，而在 Smarty 模板页中，不可以使用 SESSION 变量，那么该如何判断用户是否具有访问权限呢？这就是本实例将要讲解的内容，即应用 if 语句在 Smarty 模板页中判断用户是否具有访问权限，其运行结果如图 14.9 所示。

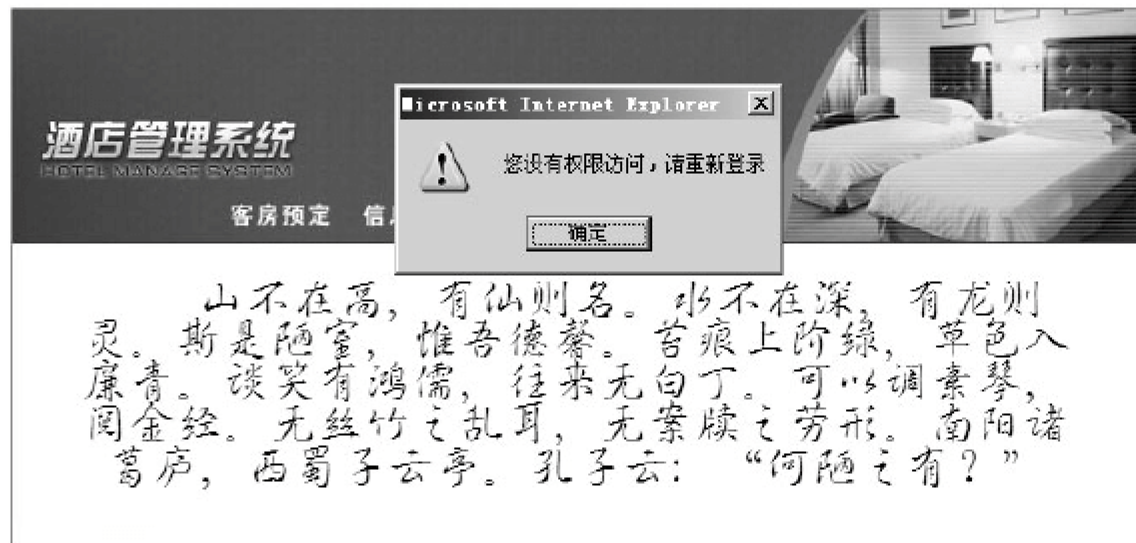


图 14.9　if 语句判断用户是否具有访问权限



实例 561　Smarty 模板中生成数字验证码

光盘位置：光盘\MR\14\561

高级

趣味指数：★★★★☆

实例说明

本实例应用 Smarty 模板中的 foreach 语句在模板页中直接生成一个数字验证码，它相当于在动态 PHP 中开发的验证码，但更加简单、实用。Smarty 模板中生成验证码的运行结果如图 14.10 所示。



图 14.10　Smarty 模板中生成数字验证码

关键技术

严格地说，这个验证码不是在 Smarty 模板页中生成的，只是通过 foreach 语句将 PHP 动态页中生成的验证码在模板页中输出。因此本实例开发的关键是 foreach()语句的应用和验证码的生成。

（1）应用 mt_rand()函数生成验证码，其语法如下：

```
int mt_rand ( [int min, int max])
```

如果 mt_rand()函数没有提供可选参数 min 和 max，则返回 0~RAND_MAX 之间的伪随机数。

（2）应用 explode()函数将生成的验证码写入到数组中，其语法如下：

```
array explode(string separator, string string, [int limit])
```

返回由字符串组成的数组，每个元素都是 string 的一个子串，它们被字符串 separator 作为边界点分隔出来。如果设置 limit 参数，则返回的数组包含最多 limit 个元素，而最后那个元素将包含 string 的剩余部分；

如果 separator 为空字符串（""），explode()函数将返回 FALSE；如果 separator 所包含的值在 string 中找不到，那么 explode()函数将返回包含 string 单个元素的数组；如果参数 limit 是负数，则返回除了最后的-limit 个元素外的所有元素。

（3）通过 assign()方法将验证码数组赋给模板变量。

（4）应用 foreach 语句循环输出模板变量中的数组元素。foreach 的语法如下：

```
{foreach name=foreach_name key=key item=item from=arr_name}
…
{/foreach}
```

参数 name 为该循环的名称；key 为当前元素的键值；item 是当前元素的变量名；from 是该循环的数组。其中，item 和 from 是必选参数，不可省略。

设计过程

（1）创建 system 文件夹，封装 Smarty 模板的配置方法，这里不再赘述。

（2）创建 index.php 文件，生成验证码，并且将验证码赋给模板变量，最后指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");                          //包含配置文件
$array= explode(' ', mt_rand(1000,9999));                       //生成随机验证码
$smarty->assign('title','Smarty 模板中生成数字验证码');          //将指定数据赋给模板变量
$smarty->assign('content',$array);                              //将数组赋给模板变量
$smarty->display('index.html');                                 //指定模板页
?>
```

（3）创建 index.html 模板文件，设计用户登录页面，添加用户登录的表单元素，同时应用 foreach 语句输出模板变量中传递的验证码，其关键代码如下：

```
<form id="form1" name="form1" method="post" action="index_ok.php">
        <input name="user" type="text" id="user" size="10" />
        <input type="hidden" name="checks" value="{foreach key=key item=item from=$content} {$item} {/foreach}" />
        <input name="check" type="text" size="8" />
        <input name="pass" type="password" id="pass" size="10" />
        <input type="image" name="imageField" src="images/Blog_03_03.jpg" />
        <input type="image" name="imageField2" onclick="form.reset();return false;" src="images/Blog_03_06.jpg" />
</form>
```

（4）创建 index_ok.php 文件，获取表单中提交的数据，对用户登录信息进行验证。如果用户名和密码正确，则跳转到 main.php 页面，否则跳转到 index.php 页面。

（5）创建 main.php 和 main.html 文件，作为网站的主页。

秘笈心法

心法领悟 561：验证码的运用。

本实例中，应用 foreach 语句输出验证码，如果要更新验证码的值，可以单击“刷新”按钮或者按 F5 键，同样也可以在地址栏中重新输入地址。

实例 562	Smarty 模板中的页面设计 光盘位置：光盘\MR\14\562	高级 趣味指数：★★★★☆

实例说明

在 Smarty 模板页中，同样可以将网页的头、尾和主文件分别进行存储，从而达到页面重用的效果，这就是本实例将要讲解的内容——在 Smarty 模板页中设计网页页面，其运行结果如图 14.11 所示。



图 14.11　Smarty 模板中的页面设计

■ 关键技术

在 Smarty 模板页中应用 include 函数载入模板页。include 函数用于在当前模板中包含其他模板，当前模板中的变量在被包含的模板中可用。include 函数的语法如下：

```
{include file="file_name " assign=" " var="   "}
```

参数 file 指定包含模板文件的名称，为必选参数；参数 assign 指定一个变量保存包含模板的输出；参数 var 传递待包含模板的本地参数，只在待包含模板中有效。

■ 设计过程

（1）创建 system 文件夹，封装 Smarty 模板的配置方法，这里不再赘述。

（2）创建 index.php 文件，应用 switch 语句根据超链接传递的变量值，包含不同的 PHP 脚本文件，同时将对应的模板页名称赋给指定的模板变量，其代码如下：

```
<?php
require("system/system.inc.php");                        //载入配置文件
switch ($_GET['caption']){                               //应用 switch 语句，进行判断
 case "reg";
        include "reg.php";                               //包含 PHP 脚本
        $smarty->assign('admin_phtml','reg.html');       //通过模板变量指定模板页
 break;
 case "log";
        include "log.php";
        $smarty->assign('admin_phtml','log.html');
 break;
 case "from";
        include "for.php";
        $smarty->assign('admin_phtml','for.html');
 break;
 case "mes";
        include "mes.php";
        $smarty->assign('admin_phtml','mes.html');
 break;
 default:
        include "for.php";
        $smarty->assign('admin_phtml','for.html');
 break;
 }
$smarty->assign("title","Smarty 模板中的页面设计--".$_GET[caption]);
$smarty->display("index.html");                          //指定模板页
?>
```

（3）创建 index.html 模板页，应用 include 函数载入页面的头文件（top.html）、尾文件（bottom.html）和主文件，其关键代码如下：

```
{$title}
{include file=top.html}
{include file=$admin_phtml}
{include file=right.html}
{include file=bottom.html}
```

（4）创建 include 函数加载的模板文件和动态 PHP 文件。

■ 秘笈心法

心法领悟 562：switch 语句的应用。

在程序设计中，所有依据条件做出判定的问题，都可以用 if 条件语句来解决。不过，在用 if...else 语句处理多个条件的判定问题时，组成条件的表达式在每一个 elseif 中都要计算一次，比较繁琐。为了避免 if 语句的冗长，提高程序的可读性，可以使用 switch 分支控制语句。switch 语句的语法如下：

```
switch(variable){
 case value1:
        statement1;
        break;
```

```
case value2:
…
default:
        default statement n;
}
```

switch 语句根据 variable 的值，依次与 case 中的 value 值相比较，如果不相等，继续查找下一个 case；如果相等，就执行对应的语句，直到 switch 语句结束或遇到 break 为止。一般 switch 语句的最后都有一个默认值 default，如果在前面的 case 中没有找到相符的条件，则输出默认语句，和 else 语句类似。

实例 563 Smarty 模板中直接定义 CSS 样式

高级

光盘位置：光盘\MR\14\563

趣味指数：★★★★☆

实例说明

由于在 Smarty 模板中默认使用“{”和“}”作为定界符，所以如果直接在 Smarty 模板中定义 CSS 样式，将输出如图 14.12 所示的错误。

该错误出现的原因是在 Smarty 模板页中直接将 CSS 样式中使用的“{”和“}”解析为 Smarty 模板的定界符，为了避免在 Smarty 模板中直接定义 CSS 样式时出现错误，Smarty 提供了特殊的处理方法，这就是本实例将要讲解的 literal 标签。

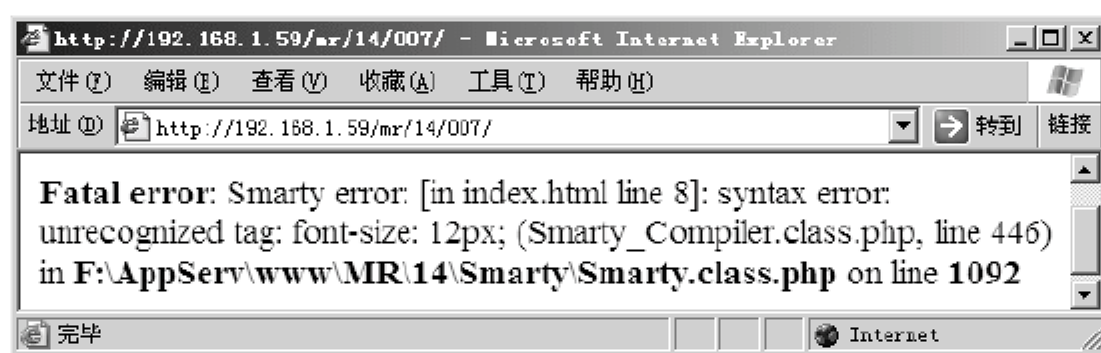


图 14.12 在 Smarty 模板中直接定义 CSS 样式时出现的错误

关键技术

literal 标签，将区域内的数据当作文本处理，此时模板将忽略其内部的所有字符信息。该特性用于显示有可能包含大括号等字符信息的 CSS 样式或者 JavaScript 脚本，当这些信息处于 {literal}{/literal} 标签中时，模板引擎将不进行分析，而直接显示。

设计过程

这里仍然应用实例 561 中的内容，唯一的区别是在 index.html 模板页中，通过{literal}{/literal} 标签定义一个 CSS 样式，控制页面中表格、文字的背景、字体和大小等，其关键代码如下：

```
{literal}
<style type="text/css">
<!--
body,td,th {
 font-size: 12px;
}
body {
 margin-left: 0px;
 margin-top: 0px;
 margin-right: 0px;
 margin-bottom: 0px;
}
a:link {
 text-decoration: none;
}
a:visited {
 text-decoration: none;
}
a:hover {
 text-decoration: none;
}
a:active {
```



设计过程

这里仍然应用实例 561 中的内容，唯一的区别是在 index.html 模板页中，直接定义 JavaScript 脚本对表单中提交的数据进行判断，通过 ldelim 和 rdelim 标签在 Smarty 模板中输出“{”和“}”，其关键代码如下：

```
<script language=javascript>
function check_form() {ldelim}
 if (form.user.value == ''){ldelim}
            alert('请输入用户名');
        form.user.focus();
        return false;
 {rdelim}
 if (form.pass.value == ''){ldelim}
            alert('请输入密码');
        form.pass.focus();
        return false;
 {rdelim}
 if (form.check.value == ''){ldelim}
            alert('请输入验证码');
        form.check.focus();
        return false;
 {rdelim}
{rdelim}
</script>
```

本实例中的其他内容这里不再赘述，其完整代码请参考本书光盘。

秘笈心法

心法领悟 564：literal 标签与 ldelim、rdelim 标签的比较。

literal 标签将“{”与“}”之间的内容按普通的文本文件来解析，控制的是内容；而 delim 与 rdelim 标签在 Smarty 模板中输出“{”与“}”，是对大括号本身进行操作。在 Smarty 模板中完成 CSS 样式或者 JavaScript 脚本的嵌入操作时，笔者个人认为还是使用 literal 标签方便一些。

实例 565	html_option 函数向下拉列表中添加列表项 光盘位置：光盘\MR\14\565	高级 趣味指数：★★★★☆

实例说明

在开发程序的过程中，经常用到下拉列表框，可以直接在页面中为其定义值，也可以使用从数据库中读取的数据来定义，但无论使用哪种方法都必须应用 option 标签，这是在常规的 Web 页面中为下拉列表赋值的方法。

本实例将介绍通过 Smarty 模板中的自定义函数 html_options 为下拉列表框赋值。使用 Smarty 模板中的自定义函数为下拉列表框赋值与使用 option 标签实现的效果是相同的，不过这种方法更加适合从数据库中读取数据，更符合 Smarty 开发模式动静分离的原则，运行结果如图 14.14 所示。

图 14.14　通过 html_options 函数为下拉列表赋值

关键技术

自定义函数 html_options 根据给定的数据创建选项组。该函数可以指定哪些元素被选定，要么指定 values 和 ouput 属性，要么指定 options 替代。自定义函数的语法如下：

```
<select name=customer_id>
 {html_options values=$cust_ids selected=$customer_id output=$cust_names}
</select>
```

html_options 函数的参数说明如表 14.1 所示。

表 14.1　html_options 函数的参数说明

参　数	类　型	说　明
name	string	下拉菜单名称，默认值为空
options	associative array	包含值和显示的关联数组
selected	string/array	已选定的元素或元素数组
output	array	包含下拉列表各元素显示值的数组
values	array	包含下拉列表各元素值的数组

如果给定值是数组，将作为 OPTGROUP 处理，且支持递归。所有的输出与 XHTML 兼容。

如果指定可选属性 name，该选项列表将被置于<select name="groupname"></select>标签中，如果没有指定，只产生选项列表。

设计过程

（1）创建 system 文件夹。首先，定义 system.smarty.inc.php 文件，封装 Smarty 的配置方法以及 ADODB 连接和操作数据库的方法。然后，定义 system.inc.php 文件，对 Smarty 配置类、数据库的连接和操作类进行实例化，并返回连接对象。最后，创建 Smarty 文件夹，定义 Smarty 的编译文件、配置文件和缓存文件的存储目录。

（2）创建 index.php 文件。首先，设置页面的编码格式、载入配置文件。然后，生成随机验证码，并且将随机验证码的值赋给模板变量。接着，从数据库中读取数据，将读取的数据存储到数组中，并且将数组赋给模板变量，作为下拉列表中的值。最后，指定模板页，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require_once("system/system.inc.php");                                  //包含配置文件
$array= explode(' ', mt_rand(1000,9999));                               //生成随机验证码
$smarty->assign('title','html_option 函数向下拉列表中添加列表项');          //将指定数据赋给模板变量
$smarty->assign('content',$array);                                      //将数组赋给模板变量
$res=$conn->execute("select * from tb_bccd ");                          //执行 select 查询语句
$array=$res->GetArray();                                                //返回查询结果
$array_id=array();
$array_name=array();
for($i=0;$i<$res->RecordCount();$i++){
        $array_id[]=$array[$i][id];
        $array_name[]=$array[$i][name];
}
$smarty->assign('cust_id', $array_id);                                  //设置下拉列表的值
$smarty->assign('cust_name', $array_name);                              //设置列表的显示数据
$smarty->display('index.html');                                         //指定模板页
?>
```

（3）创建 index.html 模板页。首先，创建 form 表单，添加表单元素，将用户的注册信息提交到 index_ok.php 文件中，其中，通过 html_options 自定义函数输出下拉列表框的值，通过 foreach 语句输出验证码的值。然后，应用 literal 标签在页面中直接嵌入 CSS 样式和 JavaScript 脚本，对用户提交的信息进行验证，其关键代码如下：

```
<form id="form" name="form" method="post" action="index_ok.php" onsubmit="return check_form(); ">
词典选择：<select name="customer" id="customer">{html_options output=$cust_name values=$cust_id }</select>
验证码：  <input type="hidden" id="checks" name="checks" value="{foreach key=key item=item from=$content} {$item} {/foreach}" />
<input name="check" type="text" id="check" size="8" />{foreach key=key item=item from=$content} {$item} {/foreach}
<input type="image" name="imageField3" src="images/reg_07.jpg" />
<input type="image" name="imageField4" onclick="form.reset();return false;" src="images/reg_09.jpg" />
</form>
```

（4）创建 index_ok.php 文件，获取表单中提交的数据，将数据添加到数据表中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require_once("system/system.inc.php");                                   //包含配置文件
if($_POST['user']!=""&&$_POST['pass']!=""&&$_POST['checks']!=""){        //判断用户名和密码是否为空
        $sql="insert into tb_user(user,pass,dates,email,type)values('".$_POST['user']."','".md5($_POST['pass'])."','".date("Y-m-d")."','".$_POST['email']."',
'".$_POST['customer']."')";
        $res=$admindb->ExecSQL($sql,$conn);                              //执行 insert 添加语句
        if($res){
                echo "<script>alert('用户注册成功！ '); window.location.href='main.php';</script>";
        }else{
                echo "<script>alert('用户注册失败！ '); window.location.href='index.php';</script>";
        }
}else{
        echo "<script>alert('用户注册信息不可为空！ '); window.location.href='index.php';</script>";
}
?>
```

（5）创建 main.php 和 main.html 文件，定义用户注册成功后跳转的页面。代码请参考本书光盘。

秘笈心法

心法领悟 565：文件编码格式设置的重要性。

在 PHP 脚本中，如果没有对文件的编码格式进行设置，那么在输出 JavaScript 脚本时，如果输出的是中文字符串，那么将输出乱码，如图 14.15 所示。



图 14.15　在 JavaScript 脚本中输出乱码

所以在 PHP 中通过 header()函数设置页面的编码格式还是十分必要的，其基本的使用方法如下：

```
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
```

实例 566　Smarty 模板制作日期、时间选择器

光盘位置：光盘\MR\14\566

高级

趣味指数：★★★☆

实例说明

在 Smarty 模板中，提供自定义的日期、时间处理函数，通过它们可以得到不同格式的日期、时间信息。在本实例中，将应用 Smarty 模板中的日期、时间选择函数，设计一个日期、时间选择器，记录用户登录的时间，其运行结果如图 14.16 所示。

图 14.16　Smarty 模板中的日期、时间选择器



连接和操作数据库的方法。然后，定义 system.inc.php 文件，对 Smarty 配置类、数据库的连接和操作类进行实例化，并返回连接对象。最后，创建 Smarty 文件夹，定义 Smarty 的编译文件、配置文件和缓存文件的存储目录。

（2）创建 index.php 文件。首先，设置页面的编码格式、载入配置文件。然后，生成随机验证码，并且将随机验证码的值赋给模板变量。最后，指定 index.html 模板页，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require_once("system/system.inc.php");                                  //包含配置文件
$array= explode(' ', mt_rand(1000,9999));                               //生成随机验证码
$smarty->assign('title','Smarty 模板制作日期、时间选择器');              //将指定数据赋给模板变量
$smarty->assign('content',$array);                                      //将数组赋给模板变量
$smarty->display('index.html');                                         //指定模板页
?>
```

（3）创建 index.html 模板页，设计用户登录的 form 表单。其中，应用 html_select_date 和 html_select_time 函数生成日期、时间下拉列表框，应用 foreach 语句输出验证码，最终将表单中的数据提交到 index_ok.php 文件中，完成用户登录的操作，其关键代码如下：

```
<form id="form" name="form" method="post" action="index_ok.php" onsubmit="return check_form(); ">
        日期：{html_select_date}
        时间：{html_select_time use_24_hours=true}
        用户名：<input name="user" type="text" id="user" size="20" />
        密码：<input name="pass" type="password" id="pass" size="20" />
        验证码：<input type="hidden" id="checks" name="checks" value="{foreach key=key item=item from=$content} {$item} {/foreach}" />
        <input name="check" type="text" id="check" size="8" />{foreach key=key item=item from=$content} {$item} {/foreach}
        <input type="image" name="imageField3" src="images/reg_07.jpg" />
        <input type="image" name="imageField4" onclick="form.reset();return false;" src="images/reg_09.jpg" />
</form>
```

（4）创建 index_ok.php 文件，获取表单中提交的用户登录信息，定义查询语句验证用户提交的用户名和密码是否正确，如果正确，则根据日期、时间选择器中提交的数据，更新用户注册信息表中的最后登录时间，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require_once("system/system.inc.php");                                  //包含配置文件
if($_POST['user']!=""&&$_POST['pass']!=""&&$_POST['checks']!=""){
        $sql="select * from tb_user where user='".$_POST['user']."' and pass='".md5($_POST['pass'])."'";
        $res=$admindb->ExecSQL($sql,$conn);                             //执行 select 查询语句
        if($res){
                $login_date=$_POST['Date_Year']."-".$_POST['Date_Month']."-".$_POST['Date_Day']."
".$_POST['Time_Hour'].":".$_POST['Time_Minute'].":".$_POST['Time_Second'];
                $sqls="update tb_user set login_date='".$login_date."' where user='".$_POST['user']."' and pass='".md5($_POST['pass'])."'";
                $rs=$admindb->ExecSQL($sqls,$conn);                     //执行 update 更新语句
                echo "<script>alert('用户登录成功！'); window.location.href='main.php';</script>";
        }else{
                echo "<script>alert('用户登录失败！'); window.location.href='index.php';</script>";
        }
}else{
        echo "<script>alert('用户登录信息不可为空！'); window.location.href='index.php';</script>";
}
?>
```

（5）创建 main.php 和 main.html 文件，作为用户登录成功的跳转页面，其代码请参考本书光盘。

秘笈心法

心法领悟 566：如何获取日期、时间下拉列表框中的值。

在 index.html 模板页中直接应用{html_select_date}和{html_select_time use_24_hours=true}定义日期、时间下拉列表框，那么应该如何获取下拉列表框中对应的值呢？方法是在 index.html 和 index.php 编写完成后，运行 index.php 文件，查看其源文件即可获取日期、时间下拉列表框对应的名称。然后，就可以在表单处理页中通过 $_POST[]方法获取日期、时间的值。

14.3　综 合 应 用

实例 567	Smarty 模板制作用户注册页面 光盘位置：光盘\MR\14\567	高级 趣味指数：★★★★☆

实例说明

在本实例中，综合运用前面讲解的方法、技术，在模板页中直接嵌入 CSS 样式和 JavaScript 脚本，通过 foreach 语句输出验证码，通过 html_options 函数定义下拉列表框的值，并且增加 Ajax 技术实现用户名的无刷新验证，开发一个完整的 Smarty 用户注册功能模块，其运行结果如图 14.17 所示。

图 14.17　Smarty 模板中的用户注册

关键技术

通过 Ajax 技术实现无刷新验证整体上可以分为 3 个步骤：

第 1 步：在模板页中，通过 script 标签调用指定的 JS 文件。创建 from 表单，添加表单元素，通过 onBlur 事件调用 JavaScript 脚本函数，完成对表单元素值的判断。

第 2 步：编写 JavaScript 脚本文件，创建自定义函数，应用 Ajax 技术完成对表单元素值的验证。

第 3 步：在模板页中，通过 div 标签输出验证结果。

其中应用到的 Ajax 技术如下：

（1）在使用 XMLHttpRequest 对象发送请求和处理响应之前，首先需要初始化该对象，由于 XMLHttpRequest 不是一个 W3C 标准，所以对于不同的浏览器，初始化的方法也是不同的。

这里使用 IE 浏览器把 XMLHttpRequest 实例化为一个 ActiveX 对象，具体方法如下：

```
var http_request = new ActiveXObject("Msxml2.XMLHTTP");
```

或者

```
var http_request = new ActiveXObject("Microsoft.XMLHTTP");
```

在上面的语法中，Msxml2.XMLHTTP 和 Microsoft.XMLHTTP 是针对 IE 浏览器的不同版本而进行设置的。

（2）open()方法。

open()方法用于设置进行异步请求目标的 URL、请求方法以及其他参数信息，语法如下：

```
open("method","URL"[,asyncFlag[,"userName"[, "password"]]])
```

在上面的语法中，method 用于指定请求的类型，一般为 get 或 post；URL 用于指定请求地址，可以使用绝对地址或者相对地址，并且可以传递查询字符串；asyncFlag 为可选参数，用于指定请求方式，同步请求为 TRUE，异步请求为 FALSE，默认情况下为 TRUE；userName 为可选参数，用于指定请求用户名，没有时可省略；password 为可选参数，用于指定请求密码，没有时可省略。

（3）send()方法。

send()方法用于向服务器发送请求。如果请求声明为异步，该方法将立即返回，否则将等到接收到响应为止，其语法如下：

```
send(content)
```

在上面的语法中，content 用于指定发送的数据，可以是 DOM 对象的实例、输入流或字符串。如果没有参数需要传递，可以设置为 null。

（4）XMLHttpRequest 对象的常用属性。

XMLHttpRequest 对象的常用属性如表 14.4 所示。

表 14.4　XMLHttpRequest 对象的常用属性

属　　性	说　　明
onreadystatechange	每个状态改变时都会触发这个事件处理器，通常会调用一个 JavaScript 函数
readyState	请求的状态，有以下 5 个取值： 0 = 未初始化 1 = 正在加载 2 = 已加载 3 = 交互中 4 = 完成
responseText	服务器的响应，表示为字符串
responseXML	服务器的响应，表示为 XML。这个对象可以解析为一个 DOM 对象
status	返回服务器的 HTTP 状态码，如： 200="成功" 202="请求被接受，但尚未成功" 400="错误的请求" 404="文件未找到" 500="内部服务器错误"
statusText	返回 HTTP 状态码对应的文本

（5）XMLHttpRequest 对象与服务器交互。

XMLHttpRequest 对象的最大用途就是不需要刷新页面即可以与服务器进行交互。可以将 Ajax 与服务器的交互分为以下 3 个步骤：

① 初始化 XMLHttpRequest 对象，关键代码如下：

```
var xmlHttp = false;                                    //定义 XMLHttpRequest 对象
try {                                                   //如果浏览器支持 XMLHttpRequest 对象，创建 ActiveXObject 对象
        xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");
} catch (e) {
        try {
                xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
        } catch (e2) {}
}
if (!xmlHttp && typeof XMLHttpRequest != "undefined") {
 try{
                xmlHttp = new XMLHttpRequest();
 }catch(e3){ xmlHttp = false;}
}
```

② 设置请求状态和返回处理函数，语法格式如下：

```
xmlobj.onreadystatechange=function_name;
```

其中 xmlobj 是 XMLHttpRequest 的对象，function_name 是用来处理请求状态和返回码的函数名（即回调函数）。

③ 发送 HTTP 请求，语法格式如下：

```
xmlobj.open(send_method,url,flag);
```

其中 xmlobj 是 XMLHttpRequest 的对象，send_method 是发送方法，可以是 GET 或者 POST，与表单中使用的方法相对应；url 是页面要调用的地址；flag 是一个标记，如果为 TRUE 则表示在等待被调用页面响应的时间内可以继续执行页面代码，反之为 FALSE。

■ 设计过程

（1）定义 Smarty 的配置方法和 ADODB 连接、操作数据库的方法，有关内容请参考实例 565，这里不再赘述。

（2）创建 index.php 文件。首先，设置页面的编码格式、载入配置文件。然后，生成随机验证码，并且将随机验证码的值赋给模板变量。接着，从数据库中读取数据，将读取的数据存储到数组中，并且将数组赋给模板变量，作为下拉列表中的值。最后，指定模板页。

（3）创建 index.html 模板页。首先，创建 form 表单，添加表单元素，将用户的注册信息提交到 index_ok.php 文件中，其中，通过 html_options 自定义函数输出下拉列表框的值，通过 foreach 语句输出验证码的值。然后，应用 literal 标签在页面中直接嵌入 CSS 样式。最后，通过 script 标签载入 JavaScript 脚本文件，对用户提交的注册信息进行判断，其关键代码如下：

```
{literal}
<style type="text/css">
body,td,th {
 font-size: 12px;
}
</style>
{/literal}
<script language="javascript" src="js/createxmlhttp.js"></script>
<script language="javascript" src="js/check_form.js"></script>
<form id="form" name="form" method="post" action="index_ok.php" onsubmit="return check_form(); ">
用户名：<input name="user" type="text" id="user" onBlur="javascript:chkname()" size="20" />
<div id="user_check"><font color="#999999">请输入用户名</font></div>
词典选择：<select name="customer" id="customer">{html_options output=$cust_name values=$cust_id }</select>
验证码：<input type="hidden" id="checks" name="checks" value="{foreach key=key item=item from=$content} {$item} {/foreach}" />
<input name="check" type="text" id="check" size="8" />{foreach key=key item=item from=$content} {$item} {/foreach}
</form>
```

（4）创建 js 脚本文件夹。首先，编写 createxmlhttp.js 文件，完成 Ajax 中 XMLHttpRequest 对象的初始化操作。然后，编写 check_form.js 文件，定义 check_form()方法对表单中提交的数据进行验证；定义 chkname()方法调用 chkname.php 文件实现用户名的无刷新验证；定义 checkregemail(emails)方法验证邮箱地址的格式是否正确；定义 checkregtel(regtel)方法验证电话号码的格式是否正确。其中 chkname()方法的语法如下：

```
function chkname(){
 if(document.getElementById('user').value==""){                                                    //判断用户名表单元素值是否为空
        document.getElementById('user_check').innerHTML="<font color=#FF0000>请输入用户名！</font>";      //如果为空则输出该内容
 }else{
        var user = document.getElementById('user').value;                                           //否则获取用户名
        var url = "chkname.php?user="+user;                                                         //定义 url
        xmlhttp.open("GET",url,true);                                                               //通过 GET 方法执行
        xmlhttp.onreadystatechange = function(){                                                    //回调函数
        if(xmlhttp.readyState == 4){                                                                //判断返回值
                var msg = xmlhttp.responseText;                                                     //获取返回值
                if(msg == '1'){                                                                     //判断如果返回值为 1
        document.getElementById('user_check').innerHTML="<font color=#FF0000>用户名被占用！</font>";      //则说明用户名被占用
                        return false;
                }else if(msg == '0'){                                                               //判断如果返回值为 0
                        document.getElementById('user_check').innerHTML="<font color=green>恭喜您，可以注册!</font>";      //用户名可用
```

```
                    }else{
                        document.getElementById('user_check').innerHTML="<font color=green>"+msg+"</font>";    //输出可用用户名
                    }
                }
            }
        xmlhttp.send(null);                                                    //执行发送
    }
}
```

（5）创建 chkname.php 文件，根据超链接中传递的参数值判断提交的用户名是否被占用，如果用户名可用，则定义变量$reback 的值为 0；否则定义变量$reback 的值为 1，代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                    //设置文件编码格式
require_once("system/system.inc.php");                                   //包含配置文件
$sqls="select name from tb_name where name='".$_GET['user']."' ";
$res=$admindb->ExecSQL($sqls,$conn);                                     //执行 select 查询语句
if($res){
    $reback = '1';                                                       //用户名被占用
}else{
    $reback = '0';                                                       //用户名可用
}
echo $reback;
?>
```

（6）创建 index_ok.php 文件，将表单中提交的数据存储到指定的数据表中。

（7）创建 main.php 和 main.html 文件，编写用户注册成功后跳转的页面。

秘笈心法

心法领悟 567：Ajax 技术中 XMLHttpRequest 对象的 readyState 属性。

在 IE 浏览器中应用 XMLHttpRequest 对象的 readyState 属性，可以书写为 readystate，而在火狐浏览器下应用 XMLHttpRequest 对象的 readyState 属性，必须书写为 readyState。所以为了保证其在不同的浏览器下都可以使用，在输出时统一使用 readyState。

实例 568	Smarty 模板制作后台管理系统主页 光盘位置：光盘\MR\14\568	高级 趣味指数：★★★★☆

实例说明

在本实例中，开发一个后台管理系统，包括管理员的登录、退出、后台管理系统中各个模块的功能展示模块，当然这些模块都只是简单的架构，没有实现具体的功能。因为开发本实例的主要目的是让读者了解如何通过 Smary 模板构建后台管理系统的主页，而非某些具体功能模块的开发。本实例首先展示后台管理的登录模块，如图 14.18 所示，登录成功后将进入到后台管理系统的主页，如图 14.19 所示。



图 14.18　后台登录



图 14.19　后台管理主页



（5）创建热点链接中链接的动态 PHP 文件和模板文件。由于篇幅所限，这里不对这部分内容进行详细讲解，读者可以参考本书光盘。

秘笈心法

心法领悟 568：Smarty 模板中的 URL 编码。

在本实例中，通过超链接传递参数值时，应用 Smarty 模板中的 escape 方法，对超链接传递的参数值进行编码，进而保护传递数据的安全。

实例 569 通过 section 循环输出数据

光盘位置：光盘\MR\14\569

高级

趣味指数：★★★★☆

实例说明

在 PHP 动态页中，可以通过 while、do…while、for 和 foreach 语句实现数据的循环输出，而在 Smarty 中，也有属于自己的循环输出语句 foreach 和 section。在本实例中，将介绍通过 section 语句完成数据的循环输出，其运行结果如图 14.20 所示。



图 14.20　Smarty 模板下载页面

关键技术

section 循环语句，用于比较复杂的数组，其语法如下：

```
{section name="sec_name" loop=$arr_name start=num step=num}
```

参数说明：

name：表示循环的名称。

loop：表示循环的数组。

start：表示循环的初始位置，如果 start=2，那么说明循环从 loop 数组的第 2 个元素开始。

step：表示步长，如果 step=2，那么循环一次后，数组的指针将向下移动两位，以此类推。

Section 循环语句读取的是存储在模板变量中的数组元素，而这个数组元素值是在动态 PHP 文件中通过调用数据库操作类中的$admindb->ExecSQL 方法获取的。

究其根源就是应用 ADODB 类库中的 GetRows()方法获取的查询结果，有关其具体的设置可以参考封装的数据库操作类 AdminDB，该类存储于 system\ system.smarty.inc.php 文件中。

设计过程

（1）本实例以实例 568 中开发的后台管理系统主页为基础，首先去除后台管理系统的登录功能，直接在



```
        <td style="padding-bottom:5px; padding-left:5px; padding-right:5px; padding-top:5px;" align="left" bgcolor="#FFFFFF">{$res[id].user}</td>
        <td style="padding-bottom:5px; padding-left:5px; padding-right:5px; padding-top:5px;" align="left" bgcolor="#FFFFFF">{$res[id].email}</td>
        <td style="padding-bottom:5px; padding-left:5px; padding-right:5px; padding-top:5px;" align="left" bgcolor="#FFFFFF">{$res[id].dates}</td>
    </tr>
{/section}
```

在上述 section 语句中，将从数组的第 2 个元素开始循环，并且数组指针一次向下移动两位。

注意，在应用 section 语句时，一定要有{/section}结束标记。

实例 570 Smarty 模板中数据的分页显示

光盘位置：光盘\MR\14\570

高级

趣味指数：★★★★☆

实例说明

本实例在应用 section 语句循环输出数据库中数据的基础上，增加一个分页功能，实现对数据库中数据的分页显示，运行结果如图 14.21 所示。

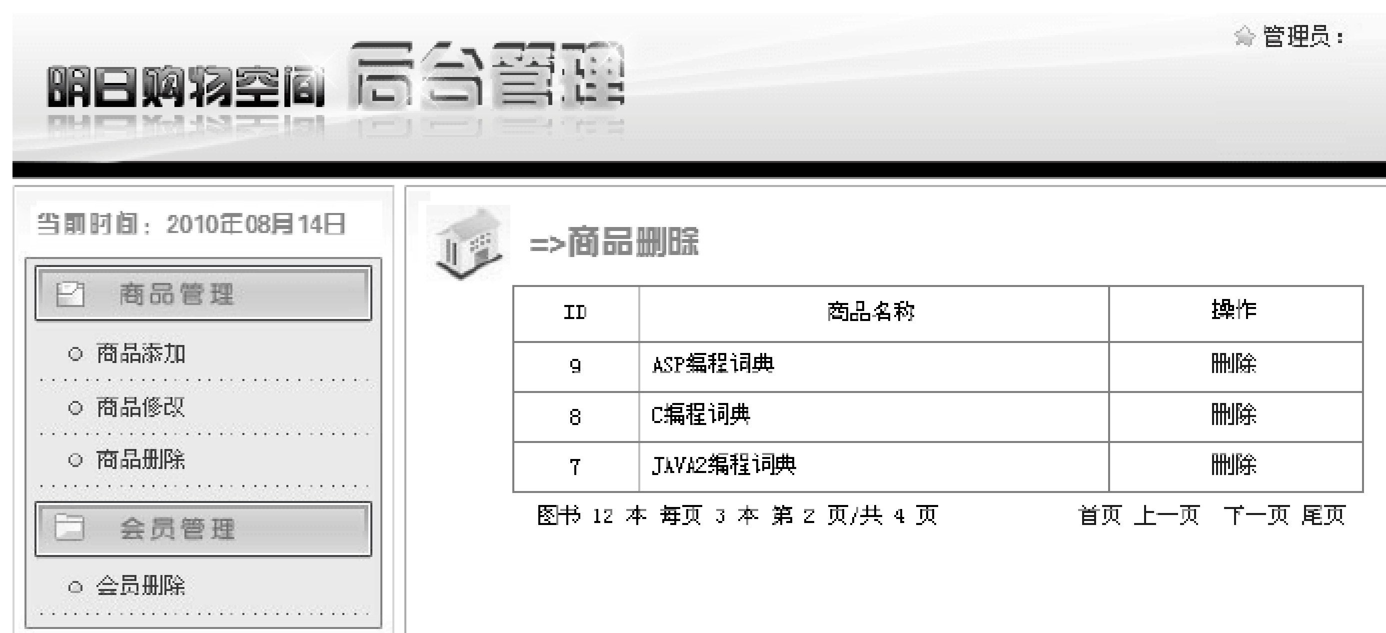


图 14.21 Smarty 模板中数据的分页输出

关键技术

在 Smarty 中实现分页与在 PHP 文件中实现分页的原理是相同的，唯一的区别是，在 Smarty 中分页的处理操作与分页的显示是分离的。分页的处理操作存储在 sho_delete.php 文件中，其应用到的关键技术如下：

（1）mysql_query()函数，执行查询、更改以及删除等操作，语法如下：

```
resource mysql_query ( string query [, resource link_identifier] )
```

参数 query 为字符串类型，指定传入的 SQL 指令；参数 link_identfier 为资源类型，指定传入的由 mysql_connect()函数或 mysq1_pconnect()函数返回的连接标识。如果省略该参数，则会使用最后一个打开的 MySQL 数据库连接。

（2）mysql_fetch_array()函数，从数组结果集中获取数据，语法如下：

```
array mysql_fetch_array ( resource result [, int result_type] )
```

参数说明：

result：资源类型的参数，要传入的是由 mysql_query()函数返回的数据指针。

result_type：可选项，整数型参数，要传入的是 MYSQL_ASSOC、MYSQL_NUM、MYSQL_BOTH 3 种由 PHP 定义的常数之一，默认值是 MYSQL_BOTH。MYSQL_ASSOC 只得到关联索引（相当于 mysql_fetch_assoc()函数）；MYSQL_NUM 只得到数字索引（相当于 mysql_fetch_row()函数）；MYSQL_BOTH 将得到一个同时包含关联和数字索引的数组。

（3）array()函数，定义一个数组，返回根据参数建立的数组。参数可以用“=>”运算符给出索引。array()函数是一个语言结构，用于字面上表示数组，不是常规的函数，语法如下：

```
array array ( [mixed ...])
```

参数 mixed 的语法为 key => value，多个参数间用逗号分开，分别定义了索引和值。

索引可以是字符串或数字。如果省略了索引，会自动产生从 0 开始的整数索引。如果索引是整数，则下一个产生的索引将是目前最大的整数索引+1。如果定义了两个完全一样的索引，则后面一个会覆盖前一个。

（4）array_push()函数，将数组当成一个栈，并将传入的变量压入该数组的末尾，该数组的长度将增加入栈变量的数目，返回数组新的单元总数。语法如下：

```
int array_push ( array array, mixed var [, mixed ...])
```

参数说明：

array：必选参数，输入的数组。

var：必选参数，用来压入数组的值。

var2：可选参数，用来压入数组的值。

（5）通过 PHP 中的函数、方法将从数据库中读取的数据进行分页处理，通过$smarty 对象调用 assign()方法将分页处理返回的变量传递给模板变量，并通过 display()方法指定模板页。

（6）在模板页 sho_delete.html 中，获取模板变量传递的值，完成数据的分页输出。

设计过程

本实例开发后台管理系统中的商品删除模块，其对应的操作文件是 sho_delete.php 和 sho_delete.html。要执行商品的删除操作，首先必须输出数据库中存储的商品数据，这是本实例的重点，再在 Smarty 模板中实现数据的分页输出。

（1）编辑 sho_delete.php 文件，连接数据库、载入 Smarty 配置文件，通过 SQL 语句读取数据库中的数据，并且定义数据的分页输出方法，应用 Smarty 中的 assign()方法将分页输出的变量值传递给模板页，最终指定模板页，其代码如下：

```
<?php
include_once "conn/conn.php";                                      //连接数据库
require_once("system/system.inc.php");                             //调用指定的配置文件
//执行查询语句，从数据库中读取商品信息
$sql=mysql_query("select count(*) as total1 from tb_bccd ",$conn);
$info=mysql_fetch_array($sql);
$total1=$info[total1];                                             //统计数据库中数据总数
if(empty($_GET[pages])==true || is_numeric($_GET[pages])==false){  //判断变量 pages 是否为空
 $page1=1;                                                         //如果变量为空，则赋值为 1
}else{                                                             //如果不为空，则获取变量的值
 $page1=intval($_GET[pages]);
}
 $pagesize1=3;                                                     //定义每页显示 3 条记录
 if($total1<$pagesize1){                                           //判断如果数据库中数据小于每页显示的记录数
        $pagecount1=1;                                             //则定义 pagecount 变量的值为 1
 }else{
        if($total1%$pagesize1==0){
                $pagecount1=intval($total1/$pagesize1);            //用总的记录数除以每页显示的记录数，获取共有几页
        }else{
                $pagecount1=intval($total1/$pagesize1)+1;
        }
 }
//将要输出的数据赋给 assign 模板变量
$smarty->assign("total1",$total1);
$smarty->assign("pagesize1",$pagesize1);
$smarty->assign("page1",$page1);
$smarty->assign("pagecount1",$pagecount1);
$query=mysql_query("select * from tb_bccd order by id desc limit ".($page1-1)*$pagesize1.",$pagesize1",$conn );
$myrow=mysql_fetch_array($query);
$array=array();                                                    //定义一个空数组
do{
    array_push($array,$myrow);                                     //将获取的数组值写入到新的数组中
}while($myrow=mysql_fetch_array($query));                          //循环读取数据库中数据
if(!$array){
        $smarty->assign("iscommo","F");                            //判断如果执行失败，则输出模板变量 iscommo 的值为 F
```

```
}else{
        $smarty->assign("iscommo","T");                    //判断如果执行成功，则输出模板变量 iscommo 的值为 T
        $smarty->assign("arr",$array);                     //定义模板变量 arr，输出数据库中的数据
}
$smarty->assign(title,"商品删除");
?>
```

（2）编辑 sho_delete.html 模板页，应用 section 语句和模板变量完成数据的分页输出，其关键代码如下：

```
<table width="545" border="1" cellpadding="1" cellspacing="1" bordercolor="#FFFFFF" bgcolor="#3399CC">
  <tr>
    <td height="25" align="center" bgcolor="#FFFFFF">ID</td>
    <td align="center" bgcolor="#FFFFFF">商品名称</td>
    <td align="center" bgcolor="#FFFFFF">操作</td>
  </tr>
  {section name=id loop=$arr}
  <tr>
    <td align="center" bgcolor="#FFFFFF">{$arr[id].id}</td>
    <td bgcolor="#FFFFFF">{$arr[id].name}</td>
    <td align="center" bgcolor="#FFFFFF"><a href="delete.php?conn_id={$arr[id].id}">删除</a></td>
  </tr>
  {/section}
</table>
<table width="545" height="25" border="0" align="center" cellpadding="0" cellspacing="0">
  <tr>
    <td>图书 {$total1} 本 每页{$pagesize1}本 第 {$page1} 页/共 {$pagecount1} 页</td>
<td align="center">{if $page1 == 1 }首页 上一页{else}<a href="index.php?caption={$title|escape:"url"}&pages=1">首页</a> 
<a href="index.php?caption={$title|escape:"url"}&pages={$page1-1}" >上一页</a>{/if} {if $page1 == $pagecount1 }下一页 尾页{else}
<a href="index.php?caption={$title|escape:"url"}&pages={$page1+1}">下一页</a> 
<a href="index.php?caption={$title|escape:"url"}&pages={$pagecount1}">尾页</a>{/if}
</td>
</tr>
</table>
```

（3）商品的删除通过 delete.php 文件，以超链接传递的参数 conn_id 为条件，执行 delete 删除语句，其代码如下：

```
<?php
include_once "conn/conn.php";                                        //连接数据库
require_once("system/system.inc.php");                               //调用指定的文件
$sql=mysql_query("delete from tb_bccd where id='".$_GET['conn_id']."'",$conn);    //执行删除语句
if($sql){
        echo "<script>alert('数据删除成功！');window.location.href='index.php';</script>";
}else{
        echo "<script>alert('数据删除失败！');window.location.href='index.php';</script>";
}
?>
```

秘笈心法

心法领悟 570：将 mysql_fetch_array()函数返回的结果在 Smarty 模板中输出。

mysql_fetch_array()函数返回的结果集是一维数组，而 Smarty 模板中 section 循环语句读取的是二维数组中的数据，所以要将 mysql_fetch_array()函数返回的结果集在模板页中循环输出，必须要将结果集添加到一个数组中，然后才能应用 section 语句，其具体的方法可以参考本实例中的 sho_delete.php 文件。



实例 571	Smarty+ADODB 完成数据的分页显示 光盘位置：光盘\MR\14\571	高级 趣味指数：★★★★

实例说明

本实例将介绍通过面向对象的方法完成 Smarty 的配置、ADODB 连接和操作 MySQL 数据库，以及分页的功能，其运行结果如图 14.22 所示。



图 14.22　Smarty+ADODB 完成数据的分页显示

关键技术

应用 Smarty+ADODB 完成数据的分页显示，涵盖了 ADODB 连接、操作 MySQL 数据库以及 ADODB 中的分页技术，而 Smarty 依旧是将 ADODB 操作 MySQL 获取到的数据通过 assign()方法传递给模板变量，然后在模板页中应用 section 语句循环输出数据。

其中有关 ADODB 连接、操作 MySQL 数据库以及 ADODB 中分页技术的详细讲解请读者参考本书第 13 章的内容，这里不再赘述。

在本实例中，将 Smarty 的配置方法以及 ADODB 连接和操作数据库的方法都存储在 system.smarty.inc.php 文件中；将 Smarty 配置类、数据库的连接和操作类的实例化存储在 system.inc.php 文件中；将 Smarty 的编译文件、配置文件和缓存文件的存储目录放置在 Smarty 文件夹下；而将 Smarty 类库和 ADODB5 类库存储在实例根目录的同级目录下。本实例的文件夹架构如图 14.23 所示。

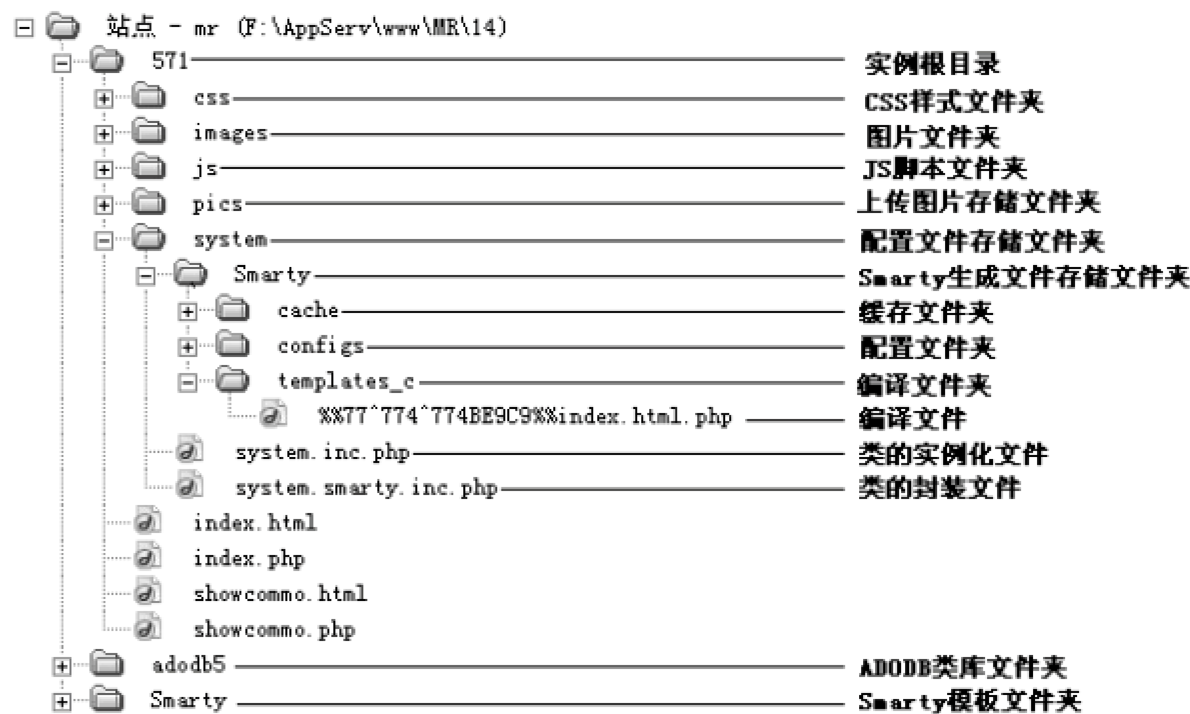


图 14.23　本实例的文件夹架构

设计过程

（1）创建 system 文件夹，存储 ADODB 类库和 Smarty 模板文件。创建 system.smarty.inc.php 文件，定义

数据库连接类、数据库操作类和分页类。

数据库连接类 ConnDB，定义构造方法 ConnDB 为成员变量赋值；定义 GetConnId()方法，应用 NewADOConnection()函数连接 MySQL、mssql 或者 Access 数据库，设置数据库编码格式为 utf-8，最后返回数据库连接对象；定义 CloseConnId()方法，关闭与数据库的连接。

数据库操作类 AdminDB，定义 ExecSQL()方法完成对数据库的添加、更新和删除操作。首先，通过 substr()函数截取 SQL 语句中前 6 个字符串，并将截取的字符串转换成小写。然后，通过 Execute()函数执行 SQL 语句。最后，根据截取的字符串判断 SQL 语句的类型，如果是 select 查询语句，则执行 GetRows()函数，如果查询结果为 0，则返回 FALSE，否则返回查询的数组；如果 SQL 语句为 update、insert 或者 delete 类型，执行成功则返回 TRUE，否则返回 FALSE。

分页类 SepPage，定义 ShowDate()方法完成从数据库中读取数据的操作，执行 PageExecute()函数实现分页功能，并且将返回值定义到数组中；定义 ShowPage()方法根据查询结果，应用分页函数创建分页超链接，并且将返回结果定义到变量$str 中。

Smarty 配置类 SmartyProject，继承 Smarty 类库中 Smarty 类，设置本实例中 Smarty 缓存文件、配置文件、模板文件和编译文件的存储位置。

system.smarty.inc.php 文件的代码请参考本书光盘，由于篇幅所限这里不再介绍。

（2）创建 system.inc.php 文件，完成对数据库连接、操作、分页以及 Smarty 配置类的实例化操作，其代码如下：

```
<?php
require("system.smarty.inc.php");                                          //包含配置类文件
$smarty=new SmartyProject();                                               //实例化配置类
$connobj=new ConnDB("mysql","localhost","root","111","db_database14",false); //实例化数据库连接类
$conn=$connobj->GetConnId();
$admindb=new AdminDB();                                                    //实例化数据库操作类
$seppage=new SepPage();                                                    //实例化分页类
?>
```

至此，有关 ADODB 和 Smarty 的安装和配置方法介绍完毕。

（3）创建 index.php 文件，调用分页类中的 ShowDate()方法，分页读取数据库中的数据，并且将返回值赋给模板变量；调用分页类中的 ShowPage()方法，完成分页超链接的输出，同样将返回值赋给模板变量，最后指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");  //包含配置文件
$arraybbs=$seppage->ShowDate("select * from tb_commo where isnew = 1 order by id ",$conn,3,$_GET["page"]);  //调用分页类，实现分页功能
if(!$arraybbs){
      $smarty->assign("isbbs","F");
}else{
      $smarty->assign("isbbs","T");
 $smarty->assign("showpage",$seppage->ShowPage("商品","个","","a1"));       //定义输出分页数据的模板变量 showpage
      $smarty->assign("arr",$arraybbs);
}
 $smarty->assign('title','Smarty+Adodb 完成数据分页显示');
 $smarty->display('index.html');
 ?>
```

（4）创建 index.html 模板文件，应用 section 语句和模板变量完成数据的分页输出。

（5）创建 showcommo.html 和 showcommo.php 文件，用于输出指定产品的详细信息。

秘笈心法

心法领悟 571：在模板页中嵌入 PHP 脚本。

在 Smarty 模板页中也可以嵌入 PHP 脚本，应用的是 Smarty 中的 PHP 标签。PHP 标签中的语句是否被处理取决于$php_handling 的设置。通常情况下是不需要在模板页中嵌入 PHP 脚本的，但是不代表不可以使用。

实例 572　Smarty 模板中日期、时间的格式化输出

光盘位置：光盘\MR\14\572

高级

趣味指数：★★★★☆

实例说明

在 PHP 脚本中，完成日期、时间的格式化输出最常用的是 date()函数，那么在 Smarty 模板中该如何完成日期、时间的输出呢？这就是本实例要讲解的内容，即通过 Smarty 模板中的 date_format 函数完成日期、时间的格式化输出，其运行结果如图 14.24 所示。



图 14.24　Smarty 模板中日期的格式化输出

关键技术

（1）Smarty 模板中的 date_format 函数，格式化从函数 strftime()获得的时间和日期，其应用示例如下：

```
{$smarty.now|date_format:"%A, %B %e, %Y":$times}
```

参数说明：

$smarty.now：传递给 date_format 的数据。这个参数可以是在 PHP 动态页中定义的日期、时间模板变量，也可以使用 Smarty 中的保留变量$smarty.now 或者 Smarty 模板中的日期。

%A, %B %e, %Y：date_format 函数执行格式化操作时使用的格式。

$times：当传递给 date_format 的数据为空时，通过$times 设置 date_format 格式化的默认值。

（2）date_format 函数可以使用的转换格式很多，其常用的转换格式说明如下：

```
%a - 根据当地格式输出“星期”缩写格式
%A - 根据当地格式输出“星期”全称格式
%b - 根据当地格式输出“月”缩写格式
%B - 根据当地格式输出“月”全称格式
%Y - 根据当地格式输出“年”全称格式
```

要了解 date_format 函数所有可以使用的转换格式，建议读者参考 Smarty 模板的参考手册。

设计过程

本实例继续使用实例 571 的内容，在 index.html 模板文件中插入 date_format 函数，输出系统的当前时间，其关键代码如下：

```
<td width="480" height="18" align="left">当前时间：  {$smarty.now|date_format:" %A   %B - %e - %Y"}</td>
<!--{$smarty.now|date_format}
{$smarty.now|date format:"%A, %B %e, %Y":$times}
{$smarty.now|date_format:"%H:%M:%S"}
{$yesterday|date_format}
```

```
{$yesterday|date_format:"%A, %B %e, %Y"}
-->
```

本实例的其他内容可以参考实例 571，这里不再赘述。

秘笈心法

心法领悟 572：date_format 函数的妙用。

通过 date_format 函数不但可以输出系统的当前时间，如果配合 strtotime()函数，还可以获取其他时间。例如，获取系统前一天的时间，其方法如下：

首先在 index.php 中，应用 strtotime()函数获取前一天的时间戳，并且将时间戳赋给模板变量。然后在 index.html 模板页中，应用 date_format 函数格式化输出前一天的时间，其关键代码如下：

```
//Index.php 文件
$smarty->assign('yesterday', strtotime('-1 day'));
//Index.html 文件
{$ yesterday |date_format:" %A    %B - %e - %Y"}
```

实例 573 Smarty 模板中的编码

光盘位置：光盘\MR\14\573

高级

趣味指数：★★★★☆

实例说明

在 PHP 脚本中应用 urlencode()函数对超链接中传递的参数进行编码，而在 Smarty 模板中提供 escape 函数对字符串进行编码。本实例将详细讲解该函数的应用，通过它对超链接中传递的参数进行编码，其运行结果如图 14.25 所示。



图 14.25　Smarty 模板中对超链接的参数值进行编码

关键技术

escape 变量用于 html 转码、url 转码，在没有转码的变量上转换单引号、十六进制转码或者 javascript 转码，默认是 html 转码，其基本的应用格式如下：

```
{$articleTitle|escape}
{$articleTitle|escape:"html"} {* escapes & " ' < > *}
{$articleTitle|escape:"htmlall"} {* escapes ALL html entities *}
{$articleTitle|escape:"url"}
{$articleTitle|escape:"quotes"}
<a href="mailto:{$EmailAddress|escape:"hex"}">{$EmailAddress|escape:"hexentity"}</a>
```

设计过程

（1）这里仍然应用实例 568 中的内容，为不同模块之间跳转的超链接传递的参数值编码，其具体的操作在 main.html 模板页中完成，关键代码如下：

```
<map name="Map" id="Map">
<area shape="rect" coords="29,41,88,62" href="main.php?caption={"商品添加"|escape:"url"}&amp;type={"商品管理"}" />
<area shape="rect" coords="30,71,91,90" href="main.php?caption={"商品修改"|escape:"url"}&amp;type={"商品管理"}" />
<area shape="rect" coords="31,99,91,118" href="main.php?caption={"商品删除"|escape:"url"}&amp;type={"商品管理"}" />
</map>
<map name="Map2" id="Map2">
<area shape="rect" coords="30,45,97,63" href="main.php?caption={"会员删除"|escape:"url"}&amp;type={"会员管理"}" />
<area shape="rect" coords="34,80,89,98" href="main.php?caption={"会员浏览"|escape:"url"}&amp;type={"会员管理"}" />
</map>
<map name="Map5" id="Map5">
<area shape="rect" coords="12,11,54,36" href="logout.php" />
</map>
<map name="Map6" id="Map6">
<area shape="rect" coords="14,12,51,36" href="main.php?caption={"商品修改"|escape:"url"}&amp;type={"商品管理"|escape:"url"}" />
</map>
```

（2）虽然通过 escape 编码后的模板变量输出时是乱码，但是在 switch 语句中仍然可以直接使用 $_GET['caption']获取超链接的值，根据这个值作出判断，在主页中输出相应模块的内容，不需要对这个参数值进行解码。

执行判断是在 main.php 文件中完成的，其关键代码如下：

```
<?php
session_start();                                              //初始化 SESSION 变量
if($_SESSION['user']!="" and $_SESSION['pass']!=""){          //判断用户名和密码是否为空
require("system/system.inc.php");                             //包含 PHP 配置文件
switch ($_GET['caption']){                                    //根据超链接传递的值进行判断
 case "商品添加";
        include "sho_insert.php";                             //包含 PHP 文件
        $smarty->assign('admin_phtml','sho_insert.html');     //将 PHP 文件对应的模板文件的名称赋给模板变量
 break;                                                       //跳出循环
//省略了部分代码
 default:
        include "sho_update.php";
        $smarty->assign('admin_phtml','sho_update.html');
 break;
}
$smarty->assign("title","后台管理系统--".$_GET['caption']);   //为模板变量赋值
$smarty->assign("caption",$_GET['caption']);                  //为模板变量赋值，输出模块名称
$smarty->assign("type",$_GET['type']);                        //为模板变量赋值，输出模块所属类别
$smarty->assign("user",$_SESSION['user']);                    //为模板变量赋值，输出登录用户名称
$smarty->display("main.html");                                //指定模板页
}else{
        echo "<script>alert('您不具备访问权限！ '); window.location.href='index.html';</script>";
}
?>
```

（3）有关本实例中 Smarty 模板的配置、数据库的连接和操作，以及用户登录模块和子模块的创建方法，请读者参考实例 568，这里不再赘述。

秘笈心法

心法领悟 573：escape 编码的重要性。

经过 escape 变量编码后的字符串，在输出时将是一堆乱码，在超链接中通过这种方法可以保证参数值的安全传递。

实例 574	Smarty 模板中应用正则表达式 光盘位置：光盘\MR\14\574	高级 趣味指数：★★★★☆

实例说明

本实例讲解 Smarty 模板中正则表达式的应用，通过该方法在 Smarty 模板页完成字符串的替换操作。本实

例以实例 569 为基础，将分页输出的会员名称 mr 替换为明日科技，运行结果如图 14.26 所示。

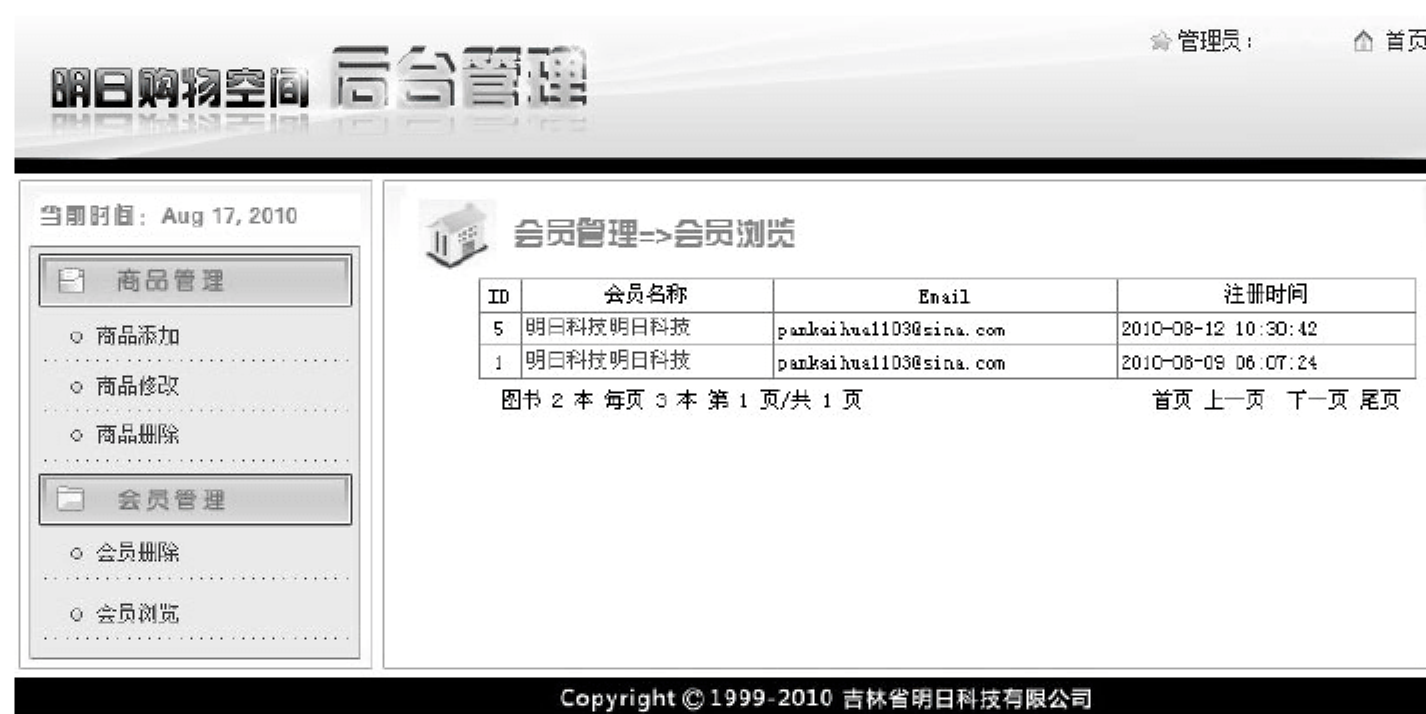


图 14.26　Smarty 模板中正则表达式的运用

关键技术

regex_replace 为 Smarty 模板中的正则表达式，可对指定的字符串进行匹配，其包括两个参数：第一个是指定的正则表达式；第二个是指定的替换文本，其应用示例如下：

```
{$name|regex_replace:"/[mr]/":"<font color='#FF0000'>明日科技</font>"}
```

在这个示例中，将模板变量$name 中的 mr 替换为明日科技。

设计过程

这里以实例 570 的内容为基础，对会员浏览模块的内容进行重新编辑，分页输出会员浏览的数据，并且应用 regex_replace 变量将会员名称中的 mr 替换为明日科技。

（1）编辑 vip_look.php 文件，分页读取数据库中存储的会员数据，并且将分页读取返回的变量赋给模板变量，最终指定模板页，其关键代码如下：

```
<?php
include_once "conn/conn.php";                                   //连接数据库
require_once("system/system.inc.php");                          //调用指定的文件
//执行查询语句，从数据库中读取商品信息
$sql=mysql_query("select count(*) as total1 from tb_name ",$conn);
$info=mysql_fetch_array($sql);
$total1=$info[total1];                                          //统计数据库中数据总数
if(empty($_GET[pages])==true || is_numeric($_GET[pages])==false){   //判断变量 pages 是否为空
        $page1=1;                                               //如果变量为空，则赋值为 1
}else{                                                          //如果不为空，则获取变量的值
        $page1=intval($_GET[pages]);
}
 $pagesize1=3;                                                  //定义每页显示 3 条记录
 if($total1<$pagesize1){                                        //判断如果数据库中数据小于每页显示的记录数
        $pagecount1=1;                                          //则定义 pagecount 变量的值为 1
 }else{
        if($total1%$pagesize1==0){
                $pagecount1=intval($total1/$pagesize1);         //用总的记录数除以每页显示的记录数，获取共有几页
        }else{
                $pagecount1=intval($total1/$pagesize1)+1;
        }
 }
//将要输出的数据赋给 assign 模板变量
$smarty->assign("total1",$total1);
$smarty->assign("pagesize1",$pagesize1);
$smarty->assign("page1",$page1);
$smarty->assign("pagecount1",$pagecount1);
$query=mysql_query("select * from tb_name order by id desc limit ".($page1-1)*$pagesize1.",$pagesize1",$conn );
$myrow=mysql_fetch_array($query);
$array=array();                                                 //定义一个空数组
```



在这个示例中，将模板变量$articleTitle 中的 Garden 替换为 Vineyard。

（2）查询关键字描红的原理：在 PHP 动态页中获取 form 表单提交的查询关键字，并且执行模糊查询，将查询结果赋给模板变量，同时将查询的关键字也赋给模板变量，然后在模板页中，通过 section 语句循环输出查询结果，并应用 replace 变量将查询结果中关键字的字体设置为红色。

设计过程

这里以实例 570 的内容为基础，对订单查询模块的内容进行重新编辑，创建 form 表单执行查询操作，并且应用 replace 变量将查询结果中的关键字描红。

（1）编辑 for_select.html 模板页，创建 form 表单，提交查询的关键字，将数据提交到 for_select.php 页面进行处理；同时在本页中应用 section 语句循环输出查询结果，并且应用 replace 对查询的关键字进行描红操作，其关键代码如下：

```
{section name=id loop=$res}
<tr>
      <td align="center" bgcolor="#FFFFFF">{$res[id].id}</td>
      <td bgcolor="#FFFFFF">{$res[id].number|replace:$select:"<font color='#FF0000'>$select</font>"}
</td>
      <td bgcolor="#FFFFFF">{$res[id].user|replace:"明日科技":"<font color='#FF0000'>明日科技</font>"}</td>
      <td bgcolor="#FFFFFF">{$res[id].dates}</td>
      <td align="center" bgcolor="#FFFFFF">{if $res[id].type==0}未处理{else}已处理{/if}</td>
</tr>
{/section}
```

（2）编辑 for_select.php 动态页，以 form 表单中提交的关键字为条件，执行模糊查询，如果查询结果为真，则将查询结果赋给模板变量，同时将查询的关键字也赋给指定的模板变量；如果查询结果为 FALSE，则为模板变量赋值 F，其代码如下：

```
<?php
$smarty->assign(title,"Smarty 模板中的关键字描红技术");                    //定义标题变量
if($_POST['number']!=""){                                                  //判断查询关键字是否为空
 $sql="select * from tb_order where number like '%".$_POST['number']."%'";  //定义模糊查询的语句
 $res=$admindb->ExecSQL($sql,$conn);                                       //执行模糊查询
 if($res){
        $smarty->assign("select",$_POST['number']);                        //将查询的关键字赋给模板变量
        $smarty->assign("boo","T");                                        //为模板变量赋值 T
        $smarty->assign("res",$res);                                       //将查询结果赋给模板变量
 }else{
        $smarty->assign("boo","F");
 }
}else{
 $smarty->assign("boo","F");
}
?>
```

秘笈心法

心法领悟 575：replace 变量的妙用。

通过 replace 变量不但可以实现查询关键字描红的功能，而且可以实现替换论坛中的非法关键字的功能，其实现方法与关键字的描红是相同的。

实例 576 Smarty 模板中控制输出字符串的行宽

光盘位置：光盘\MR\14\576

高级

趣味指数：★★★★☆

实例说明

在 Web 页面中，通常应用 CSS 样式来控制字符串输出的行宽，而在 Smarty 模板中却有一个变量可以完成

这个功能，即 wordwrap。在本实例中将讲解这个变量的应用，并且通过它控制页面中输出字符串的行宽，其运行结果如图 14.28 所示。



图 14.28　每行显示 30 个字符串的运行效果

关键技术

wordwrap 指定段落的宽度（即控制一行显示多少个字符，如果超过这个字符数则换行），其默认值是 80 字符。该方法包含 3 个参数：

第 1 个参数设置每行显示的字符数；

第 2 个参数设置在约束点使用什么字符（默认值是换行符\n）；

第 3 个参数控制 Smarty 是截取到字符串的末尾，还是精确到指定长度的字符，默认情况是截取到字符串的默认值，如果设置第 3 个参数的值为 TRUE，那么就精确截取到指定长度的字符，其应用的示例如下：

```
{$articleTitle|wordwrap:30}
```

设置段落一行中包括 30 个字符，并使用\n 进行换行操作。

```
{$articleTitle|wordwrap:30:"<br>\n"}
```

设置段落一行中包括 30 个字符，并使用
\n 进行换行操作。

```
{$articleTitle|wordwrap:30:"\n":true}
```

设置段落一行中包括 30 个字符，并使用\n 进行换行操作，同时精确截取 30 个字符。

上述示例是对模板变量$articleTitle 中存储的数据进行操作。

设计过程

（1）创建 system 文件夹。首先，定义 system.smarty.inc.php 文件，封装 Smarty 的配置方法。然后，定义 system.inc.php 文件，对 Smarty 配置类进行实例化，并返回连接对象。最后，创建 Smarty 文件夹，定义 Smarty 的编译文件、配置文件和缓存文件的存储目录。

（2）创建 index.php 文件。首先，设置页面的编码格式、载入配置文件。然后，通过 file_get_contents()函数读取存储在文本文件 content.txt 中的字符串，并应用 iconv()函数将字符串的 gb2312 编码转换为 utf-8 编码，同时将转换后的文件存储到模板变量中，最终指定模板页，代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );              //设置文件编码格式
include("system/system.inc.php");                                  //包含配置文件
$smarty->assign('title','Smarty 模板中控制输出字符串的行宽');      //定义实例标题变量
$str = file_get_contents('files/content.txt');                     //读取文本文件中的数据
```

```
$smarty->assign('content',iconv("gb2312","utf-8",$str));          //将文本文件中的数据存储到模板变量中
$smarty->display('index.html');                                    //指定模板页
?>
```

（3）创建 index.html 模板页，通过模板变量输出文本文件中的字符串，并且应用 wordwrap 控制每行显示 30 个字符，应用
\n 完成换行操作，其关键代码如下：

```
{$content|wordwrap:30:"<br>\n"}
```

秘笈心法

心法领悟 576：Smarty 模板中换行符的转换。

在应用 wordwrap 控制段落中每行显示的字符时，默认使用\n 为换行符。但是，使用\n 作为换行符，在页面中看不到字符串换行的效果，只有在查看源文件时才能看出字符串已经换行，所以，在应用 wordwrap 控制段落中每行显示的字符时，应该设置第 2 个参数，指定
为换行符，这样在页面中就可以直接看到段落换行的效果。

在 Smarty 模板中，还可以通过 nl2br 将所有的换行符转换成</br>，其使用方法与 PHP 中的 nl2br()函数相同。而本实例中的字符串换行操作也可以使用如下方法来完成：

```
{$content|wordwrap:30|nl2br}
```

实例 577 Register_object()方法注册模板对象

光盘位置：光盘\MR\14\577

高级

趣味指数：★★★★☆

实例说明

在 PHP 中，可以通过 function 创建自定义函数，完成一些特殊的操作，同样在 Smarty 中可以通过 register_object()方法注册对象，在模板页中完成一些特殊的操作。在本实例中通过 register_object()方法注册对象，定义方法 moneyFormat()和 unHtml()，分别对留言本中的评分进行转义（保留两位小数）及对留言内容中的 html 标记进行转义，其运行结果如图 14.29 所示。



图 14.29 评分和 html 标记转义后输出

关键技术

（1）register_object()方法，注册一个在模板中使用的对象，语法如下：

```
void register_object (string object_name, object $object, array allowed methods/properties, boolean format, array block methods)
```

register_object 注册对象的参数说明如表 14.5 所示。

表 14.5　register_object 注册对象的参数说明

参　　数	说　　明
object_name	模板页中使用的对象名称
$object	实例化的对象名称
allowed methods/properties	定义允许使用的方法，如果设置为 null，表示所有方法都可用
format	设置对象参数传递的方法，如果值为 false，则表示一次传递一个参数
block methods	该参数在 format 的值为 true 应用，向对象中添加一个块方法

对象注册成功后，就可以在模板页中通过对象名称调用对象中的方法，完成对指定数据的操作。本实例在模板页中通过 Util 对象分别调用 moneyFormat()和 unHtml()方法，完成对留言本中评分和内容的转义，其关键代码如下：

```
{if $iscommo=="T"}
<table width="500" border="0" align="center" cellpadding="0" cellspacing="1" bgcolor="#0099CC">
      {section name=id loop=$array}
   <tr>
           <td>{Util->unHtml pt=$array[id].id }
{Util->unHtml pt=$array[id].title }
评分：{Util->moneyFormat pt=$array[id].integral}
<br>{Util->unHtml pt=$array[id].content}
</td>
    </tr>
{/section}
</table>
{/if}
```

（2）str_replace()函数，实现字符串的替换，其语法如下：

```
mixed str_replace ( mixed search, mixed replace, mixed subject , int &count )
```

str_replace()函数将所有在参数 subject 中出现的 search 以参数 replace 替换，参数&count 表示替换字符串执行的次数。str_replace()函数的参数说明如表 14.6 所示。

表 14.6　str_replace()函数的参数说明

参　　数	说　　明
search	指定将要被替换的字符
replace	指定替换所使用的字符
subject	指定被操作的字符串
&count	替换字符串执行的次数

（3）number_format()函数，将数字字符串格式化，其语法如下：

```
string number  format ( float number , int decimals)
string number_format ( float number, int decimals, string dec_point, string thousands_sep)
```

number_format()函数返回参数 number 格式化后的字符串，该函数可以有 1 个、2 个或者 4 个参数，但不能是 3 个参数。

如果只有 1 个参数，number 格式化后会舍去小数点后的值，且每一千就会以逗号“,”来隔开；

如果有 2 个参数，number 格式化后会到小数点第 decimals 位，且每一千就会以逗号“,”来隔开；

如果有 4 个参数，number 格式化后会到小数点第 decimals 位，dec_point 用来替代小数点“.”，thousands_sep 用来替代每一千隔开的逗号“,”。

■ 设计过程

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录；创建类文件 system.smarty.inc.php，

定义数据库连接、操作、分页和 Smarty 的配置类，定义类的实例化文件 system.inc.php，完成各个类的实例化操作，并返回操作对象。

（2）创建 index.php 文件，包含类的实例化操作文件；创建类，定义评分转义和 html 标记转义的方法，通过 register_object 注册模板对象；完成数据库中数据的分页读取，最终指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");                          //调用指定的文件
class Util {
	function moneyFormat($integral) {                          //将数字转换成有两位小数的形式
		return str_replace ( ',', '', number_format ( $integral, 2 ) );
	}
	function unHtml($text) {                                   //转换字符串中的 html 标签
		$str = htmlspecialchars ( $text );
		$str = ereg_replace ( '<br>', '\n', $str );
		$str = nl2br ( $str );
		$str = ereg_replace ( ' ', ' ', $str );
		return $str;
	}
}
$ntil=new Util();                                               //实例化对象
$smarty->register_object("Util", $ntil,null,false);              //注册模板对象
$array=$seppage->ShowDate("select * from tb_guestbook order by id desc",$conn,2,$_GET["page"]);    //调用分页类，实现分页
if(!$array){
	$smarty->assign("iscommo","F");                            //判断如果执行失败，则输出模板变量 iscommo 的值为 F
}else{
	$smarty->assign("iscommo","T");                            //判断如果执行成功，则输出模板变量 iscommo 的值为 T
	$smarty->assign("showpage",$seppage->ShowPage("留言","条","","a1"));    //定义输出分页数据的模板变量 showpage
	$smarty->assign("array",$array);
}
 $smarty->assign('title','register_object()方法注册模板对象');
 $smarty->display('index.html');
?>
```

（3）创建 index.html 模板页，分页输出模板变量中传递的数据，并且调用注册模板对象中的方法，对评分和 html 标记进行转换操作。同时创建一个 form 表单，用于提交留言信息，将数据提交到 index_ok.php 文件中。关键代码请参考关键技术中的内容。

（4）创建 index_ok.php 文件，将表单中提交的留言信息存储到指定的数据表中，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );            //设置文件编码格式
require_once("system/system.inc.php");                          //调用指定的文件
$title=$_POST[title];                                           //获取标题
$content=$_POST[content];                                       //获取内容
$integral=$_POST[integral];
$createtime=date("Y-m-j H:i:s");                                //定义时间
$arr=$admindb->ExecSQL("insert into tb_guestbook(title,content,createtime,integral) values ('$title','$content','$createtime','$integral')",$conn);
if($arr){
	echo "<script>alert('留言发表成功!');window.location.href='index.php';</script>";
}else{
	echo "<script>alert('留言发表失败!');history.back();</script>";
}
?>
```

秘笈心法

心法领悟 577：模板对象的注销。

在 Smarty 模板中，不但可以注册模板对象，而且可以注销模板对象，注销模板对象应用的是 unregister_object()方法。

实例 578	Register_function()方法注册模板函数 光盘位置：光盘\MR\14\578	高级 趣味指数：★★★★☆

实例说明

本实例讲解在 Smarty 中注册模板函数的方法。模板函数可以实现与 PHP 中自定义函数相同的功能，其特点是注册在动态页中完成，实际的应用在静态页中进行，完全体现 Smarty 的动静分离的开发模式。本实例分页输出论坛帖子的信息，如果帖子内容超出指定的长度，就通过注册模板函数对其进行截取，并且使用省略号替代被截取部分，其运行结果如图 14.30 所示。



图 14.30　注册模板函数截取字符串

关键技术

（1）register_function()方法，动态注册模板函数插件，语法如下：

```
void register_function (string name, mixed impl, bool cacheable, array or null cache_attrs)
```

register_function()方法的参数说明如表 14.7 所示。

表 14.7　register_function 方法的参数说明

参　　数	说　　明
name	模板函数的名称
impl	执行函数的名称。执行函数的格式可以是一个包含函数名称的字符串；也可以是一个 array(&$object, $method)数组形式，其中&$object 是一个对象的引用，而$method 是对象的一个方法；还可以是一个 array(&$class, $method)数组形式，其中$class 是一个类的名称，$method 是类中的一个方法
cacheable	可选参数，大多数情况下可以省略
cache_attrs	可选参数，大多数情况下可以省略

（2）extract()函数，从数组中将变量导入到当前的符号表中，语法如下：

```
int extract ( array var_array [, int extract_type [, string prefix]] )
```

本函数用来将变量从数组中导入到当前的符号表中。接受结合数组 var_array 作为参数并将键名当作变量名，值作为变量的值。对每个键 / 值对都会在当前的符号表中建立变量，并受到 extract_type 和 prefix 参数的影响。

函数 extract()检查每个键名看是否可以作为一个合法的变量名，同时也检查和符号表中已有的变量名的冲突。处理非法数字和冲突的键名的方法将根据 extract_type 参数决定，可以是以下值之一：

EXTR_OVERWRITE：如果有冲突，覆盖已有的变量。



```
$smarty->register_function("Util", "unHtml");                                        //注册模板函数
$array=$seppage->ShowDate("select * from tb_guestbook order by id desc",$conn,6,$_GET["page"]);          //调用分页类，实现分页
if(!$array){
        $smarty->assign("iscommo","F");                                        //判断如果执行失败，则输出模板变量 iscommo 的值为 F
}else{
        $smarty->assign("iscommo","T");                                        //判断如果执行成功，则输出模板变量 iscommo 的值为 T
        $smarty->assign("showpage",$seppage->ShowPage("帖子","条","","a1"));          //定义输出分页数据的模板变量 showpage
     $smarty->assign("array",$array);
}
 $smarty->assign('title','register_function()方法注册模板函数');
 $smarty->display('index.html');

?>
```

（3）创建 index.html 模板页，获取模板变量传递的数据，实现数据的分页输出，并且调用动态页中注册的模板函数实现对论坛中内容的截取操作，其关键代码如下：

```
{section name=id loop=$array}
<tr>
<td colspan="4" bgcolor="#FFFFFF">{$array[id].id }  {Util text=$array[id].title }积分：{$array[id].integral}    
时间：{$array[id].createtime}<br>{Util text=$array[id].content}</td>
</tr>
{/section}
```

秘笈心法

心法领悟 578：字符串的截取。

通常使用 substr()函数对字符串进行截取，但是该函数在截取中文字符串时，很可能导致截取的中文字符串不完整，出现乱码的情况。而本实例中应用的 mb_substr()函数，同样可以实现字符串的截取功能，且可以对中文字符串进行截取，不会出现乱码的问题。

实例 579	Smarty 模板中 truncate 方法截取字符串 光盘位置：光盘\MR\14\579	高级 趣味指数：★★★★

实例说明

在 Smarty 中，提供了 truncate 方法对字符串进行截取，该方法可以截取到一个词的末尾。在本实例中，应用 truncate 方法对论坛中标题和内容进行截取，并且用省略号替换截取的内容，其运行结果如图 14.31 所示。



图 14.31　truncate 方法截取字符串

关键技术

truncate 方法，从字符串开始处截取指定长度的字符，默认是 80 个字符。应用示例如下：

```
{$articleTitle|truncate}                    <!--截取默认长度的字符串-->
```

```
{$articleTitle|truncate:30}              <!--截取 30 个字符-->
{$articleTitle|truncate:30:""}           <!--截取 30 个字符，结尾处默认使用省略号进行添加-->
{$articleTitle|truncate:30:"---"}        <!--截取 30 个字符，结尾处以“---”进行添加-->
{$articleTitle|truncate:30:"":true}      <!--截取 30 个字符，进行精确截取-->
{$articleTitle|truncate:30:"...":true}   <!--精确截取 30 个字符，以省略号进行添加-->
```

其中$articleTitle 为模板变量。

truncate 方法的参数说明如表 14.9 所示。

表 14.9　truncate 方法的参数说明

参　　数	说　　明
1	第 1 个参数，指定截取字符的数量，类型为 Integer
2	第 2 个参数，指定截取后追加在截取词后面的字符串，类型为 String
3	第 3 个参数，类型为 Boolean，如果该值为 FALSE，表示截取到词的边界；如果该值为 TRUE，表示截取时精确到指定字符

设计过程

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录；创建类文件 system.smarty.inc.php，定义数据库连接、操作、分页和 Smarty 的配置类，定义类的实例化文件 system.inc.php，完成各个类的实例化操作，并返回操作对象。

（2）创建 index.php 文件，载入类的实例化文件；分页读取数据库中存储的论坛数据，并且将读取的结果存储到模板变量中，最终指定模板页，其代码如下：

```
<?php
require_once("system/system.inc.php");                                          //调用指定的文件
$array=$seppage->ShowDate("select * from tb_guestbook order by id desc",$conn,6,$_GET["page"]);   //调用分页类，实现分页
if(!$array){
 $smarty->assign("iscommo","F");                                                //判断如果执行失败，则输出模板变量 iscommo 的值为 F
}else{
 $smarty->assign("iscommo","T");                                                //判断如果执行成功，则输出模板变量 iscommo 的值为 T
 $smarty->assign("showpage",$seppage->ShowPage("帖子","条","","a1"));  //定义输出分页数据的模板变量 showpage
     $smarty->assign("array",$array);
}
 $smarty->assign('title','truncate 方法截取字符串');                             //定义标题
 $smarty->display('index.html');                                                //指定模板页
?>
```

（3）创建 index.html 模板页，获取模板变量传递的数据，实现数据的分页输出，并且通过 truncate 方法对输出的标题和内容进行截取操作：标题截取 30 个字节，内容截取 60 个字节，应用省略号补齐截取部分，其关键代码如下：

```
{section name=id loop=$array}
<tr>
<td>
{$array[id].id} {$array[id].title|truncate:30:"...":false};
积分：{$array[id].integral}
时间：{$array[id].createtime|truncate:12:""}<br>
        {$array[id].content|truncate:60:"...":false}
</td>
</tr>
{/section}
```

秘笈心法

心法领悟 579：解决用 truncate 方法截取中文字符串时出现的乱码问题。

定位到 Smarty 类库的 plugins\modifier.truncate.php 文件中，重新编辑 smarty_modifier_truncate 方法，将其中截取字符串使用的 substr()函数全部替换为 mb_substr()函数。这样在调用 truncate 方法截取中文字符串时就不会出现乱码问题。

实例 580　开启网站注册页面的缓存

光盘位置：光盘\MR\14\580

高级

趣味指数：★★★★☆

实例说明

缓存的合理应用，应该以实际的开发需要为依据，对于那些需要经常更新的程序是否开启缓存，如果开启缓存，周期长短设置为多少，这些都必须根据程序的实际需求进行设置。在本实例中，开启网站注册页面的缓存，注册页面的运行结果如图 14.32 所示。

图 14.32　注册页面的运行结果

关键技术

缓存的设置，既要考虑可以提高网站的访问速度，又要考虑缓存生命周期的合理性。不能偏执一方，应该采取中庸之道，达到最理想的效果。

例如，网站首页是不需要经常更新的内容，缓存的生存周期就可以设置的长一些，而类似于论坛中帖子的数据，则可以不开启缓存，因为帖子的数据会不断地更新，如果使用缓存可能会导致浏览不到最新的数据。

（1）开启缓存。

开启缓存的方法非常简单，只要将 Smarty 对象中$config 的值设置为 TRUE 即可，同时还要通过 Smarty 对象中的$cache_dir 属性指定缓存文件的存储位置，其操作代码如下：

```
$smarty->caching=true;                                          //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';            //定义缓存文件存储位置
```

（2）设置缓存生命周期。

缓存创建成功后，必须为其设置一个生命周期，如果它一直不更新，那么就没有任何意义。设置缓存生命周期应用的是 Smarty 对象中的$cache_lifetime 属性，缓存时间以秒为单位，默认值是 3600 秒，其操作代码如下：

```
$smarty->caching=true;                                     //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';       //定义缓存文件存储位置
$smarty->cache_lifetime=3600                               //设置缓存时间为 1 小时
```

（3）判断模板文件是否已经被缓存。

如果页面已经被缓存，那么就可以直接调用缓存文件，而不再执行动态获取数据和输出的操作。为了避免在开启缓存后，再次执行动态获取数据和输出操作给服务器带来的压力，最佳的方法就是应用 Smarty 对象中的

is_cached()方法，判断指定的模板是否存在缓存，如果存在，则直接执行缓存中的文件，否则执行动态获取数据和输出的操作。操作代码如下：

```
$smarty->caching=true;                                    //开启缓存
if(!$smarty->is_cached('index.html')){
    //执行动态获取数据和输出的操作
}
$smarty->display('index.html');
```

（4）清除模板中缓存。

缓存的清除有两种方法：

① clear_all_cache()方法，清除所有模板缓存，其语法如下：

```
void clear_all_cache (int expire time)
```

可选参数 expire time，可以指定一个以秒为单位的最小时间，超过这个时间的缓存都将被清除。

② clear_cache()方法，清除指定模板的缓存，其语法如下：

```
void clear_cache (string template [, string cache id [, string compile id [, int expire time]]])
```

如果这个模板有多个缓存，可以用第 2 个参数指定要清除缓存的缓存号，还可以通过第 3 个参数指定编译号。可以把模板分组，以便方便地清除一组缓存。第 4 个参数是可选的，用来指定超过某一时间（以秒为单位）的缓存才会被清除。

设计过程

这里在实例 567 的基础上，完成用户注册的功能，并且开启用户注册页面的缓存。有关缓存的设置在 index.php 动态页中完成，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );            //设置文件编码格式
require_once("system/system.inc.php");                          //包含配置文件
$smarty->caching=true;                                          //开启缓存
$smarty->cache_lifetime=3600;                                   //设置缓存过期时间
//$smarty->clear_all_cache();                                   //清除所有缓存
if(!$smarty->is_cached('index.html')){
$res=$conn->execute("select * from tb_bccd ");                  //执行 select 查询语句
$array=$res->GetArray();
$array_id=array();
$array_name=array();
for($i=0;$i<$res->RecordCount();$i++){
 $array_id[]=$array[$i][id];
 $array_name[]=$array[$i][name];
}
$smarty->assign('cust_id', $array_id);                          //设置下拉列表的值
$smarty->assign('cust_name', $array_name);                      //设置列表的显示数据
}
$array= explode(' ', mt_rand(1000,9999));                       //生成随机验证码
$smarty->assign('title','开启网站注册页面的缓存');               //将指定数据赋给模板变量
$smarty->assign('content',$array);                              //将数组赋给模板变量
$smarty->assign('contents',$arrays);                            //将数组赋给模板变量
$smarty->display('index.html');                                 //指定模板页
?>
```

缓存文件存储在 system\Smarty\cache 目录下，而存储路径的设置是在 system\system.smarty.inc.php 文件中完成的。有关用户注册的其他功能，请读者参考实例 567，这里不再赘述。

秘笈心法

心法领悟 580：同一模板生成多个缓存。

在实际的程序开发中，经常会遇到同一个模板文件生成多个页面的情况。此时要对这多个页面进行缓存，就要应用 Smarty 中的 display()方法，通过该方法的第 2 个参数设置缓存号，有几个不同的缓存号就有几个缓存页面，操作代码如下：

```
$smarty->caching=true;                                          //开启缓存
$smarty->cache_dir = BASE_PATH.SMARTY_PATH.'cache/';            //定义缓存文件存储位置
```

```
$smarty->cache_lifetime=3600;                          //设置缓存时间为 1 小时
$smarty->display('index.html',$_GET['id']);            //将 id 作为第 2 个参数传递
```

实例 581　通过配置文件定义变量　高级

光盘位置：光盘\MR\14\581　趣味指数：★★★★☆

实例说明

配置文件的应用，有利于设计者管理文件中的模板全局变量。例如，定义一个模板色彩变量，一般情况下如果想改变一个程序的外观色彩，必须更改每一个文件的颜色变量。如果有配置文件，色彩变量就可以保存在一个单独的文件中，只要改变配置文件就可以实现色彩的更新，这与在 Web 页面开发中应用的 CSS 样式非常相似。在本实例中，将讲解如何应用 Smarty 中的配置文件，并通过配置文件定义页面中 body 标签的样式，其运行结果如图 14.33 所示。



图 14.33　通过配置文件定义页面的样式

关键技术

（1）创建配置文件。

配置文件可以任意命名，其存储位置由 Smarty 对象的$config_dir 属性指定。如果存在不只在一个区域内使用的变量值，可以使用三引号（"""）将其完整地封装起来。在创建配置文件时，建议在程序运行前使用“#”加一些注释信息，这样有助于程序的阅读、更新。

在配置文件中既可以声明全局变量，也可以声明局部变量。如果声明局部变量，可以使用中括号“[]”括起来，在中括号之内声明的变量属于局部变量，而中括号之外声明的变量都是全局变量。中括号的使用不仅使配置文件中声明变量的模块变得清晰，而且可以在模板中选择加载中括号内的变量。

（2）加载配置文件。

加载配置文件应用 Smarty 的内建函数 config_load，其语法如下：

```
{config_load file="file_name " section="add_attribute" scope="" global=""}
```

参数说明：

file：指定包含的配置文件的名称。

section：附加属性，当配置文件中包含多个部分时应用，指定具体从哪一部分中取得变量。

scope：加载数据的作用域，取值必须为 local、parent 或 global。local 说明该变量的作用域为当前模板；parent 说明该变量的作用域为当前模板和当前模板的父模板（调用当前模板的模板）；global 说明该变量的作用域为所有模板。

global：说明加载的变量是否全局可见，等同于 scope=parent。

提示：当指定 scope 属性时，可以设置 global 属性，但模板忽略该属性值，而以 scope 属性为准。

（3）引用配置文件中的变量。

配置文件加载成功后，就可以在模板中引用配置文件中声明的变量。引用配置文件应用的是“#”或者 Smarty 的保留变量$smarty.config，其应用示例如下：

```
{ config_load file="file_con.conf"}                        {* 加载配置文件 *}
{#title#}
<td height="228" colspan="2" align="left" valign="top" class="{$smarty.config.styles}">
```

设计过程

（1）创建 system 文件夹，定义 Smarty 文件夹，存储编译目录、缓存目录和配置文件目录；创建类文件 system.smarty.inc.php，定义 Smarty 的配置类，在该类中指定配置文件存储的目录，定义类的实例化文件 system.inc.php，完成 Smarty 类的实例化操作，并返回操作对象。

（2）创建 index.php 文件，载入类的实例化文件；通过文件系统函数 file_get_contents()读取文本文件中的数据，并且将数据存储到模板变量中，最终指定模板页，其代码如下：

```
<?php
header ( "Content-type: text/html; charset=UTF-8" );                //设置文件编码格式
include("system/system.inc.php");                                    //包含配置文件
$str = file_get_contents('files/content.txt');                       //读取文本文件中的数据
$smarty->assign('content',iconv("gb2312","utf-8",$str));             //将文本文件中的数据存储到模板变量中
$smarty->display('index.html');                                     //指定模板页
?>
```

（3）创建 index.html 模板页。首先，通过 config_load()函数加载配置文件。然后，应用“#”和$smarty.config 引用配置文件中的变量。最后，通过模板变量输出文本文件中的数据，其关键代码如下：

```
{ config_load file="file_con.conf"}                        {* 加载配置文件 *}
<title>{#title#}</title>
<body leftmargin="{#leftmargin#}" topmargin="{#topmargin#}" marginwidth="{#marginwidth#}" marginheight="{#marginheight#}">
<td height="228" colspan="2" align="left" valign="top" class="{$smarty.config.styles}">{$content}</td>
```

秘笈心法

心法领悟 581：Smarty 模板的最佳配置方案。

答案就在本章的实例中，应用面向对象技术，将配置方法封装到类中，应用 Smary 模板时调用类的实例化文件即可。通过这种方法配置 Smarty 模板，不必再设置服务器的绝对路径和 Smarty 类库的绝对路径，非常简单、方便。



第15章 综合应用

- PHP 页面编码
- 网站公共文件设计
- 注册、登录
- 数据的输出
- 站内搜索
- 高级搜索
- 购物车
- 图书订单处理
- 后台管理系统登录
- 系统管理
- 图书类别管理
- 图书管理
- 用户管理
- 订单信息管理
- 通用信息管理

15.1　PHP 页面编码

PHP 文件的页面编码是指文件以何种形式保存，其中支持简体中文的编码主要有 GB2312、GBK 和 utf-8 等，本节将对 PHP 常用的页面编码格式和 PHP 的国际化通过具体实例进行讲解。

实例 582	设计 GB2312 编码格式的网页 光盘位置：光盘\MR\15\582	中级 趣味指数：★★★

实例说明

本实例主要讲解如何设置页面编码格式为 GB2312 编码，运行本实例，如图 15.1 所示，在页面中打印出明日企业网的页面编码的提示信息，从运行结果可以知道该页面所使用的编码为 GB2312 编码。

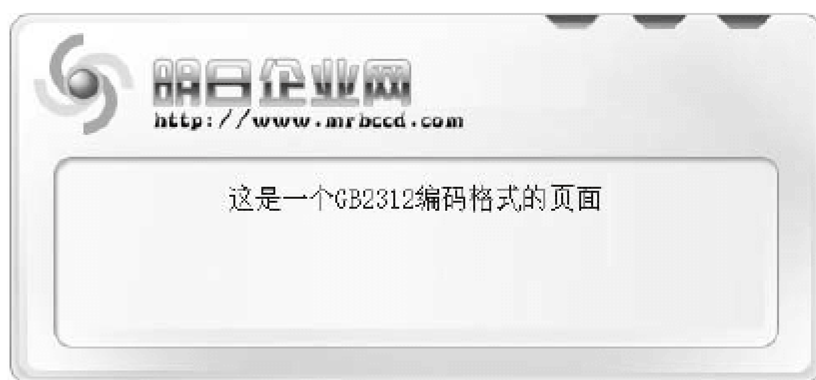


图 15.1　打印明日企业网的页面编码

关键技术

设计 GB2312 编码格式的网页需要进行两点设置，首先使用编辑器设置页面文件的编码为 GB2312 格式，然后通过 HTML 语言的 meta 标记或 PHP 的 header()函数设置页面编码为 GB2312。

设置页面文件编码可以使用 UltraEdit、Edit Plus 和 Zend Studio 等编辑器。下面将以 Zend Studio 为例讲解如何设置页面文件编码。

打开 Zend Studio 后，选择要设置编码的文件，然后单击右键，将弹出如图 15.2 所示的页面，选择“属性”选项，将弹出如图 15.3 所示的页面属性对话框。

在页面属性对话框中选择文件编码模块，在该模块的其他编码的下拉列表框中选择 GBK 编码，然后单击“应用”按钮，最后单击“确定”按钮即可完成对页面文件编码的设置。



图 15.2　页面常规操作下拉列表

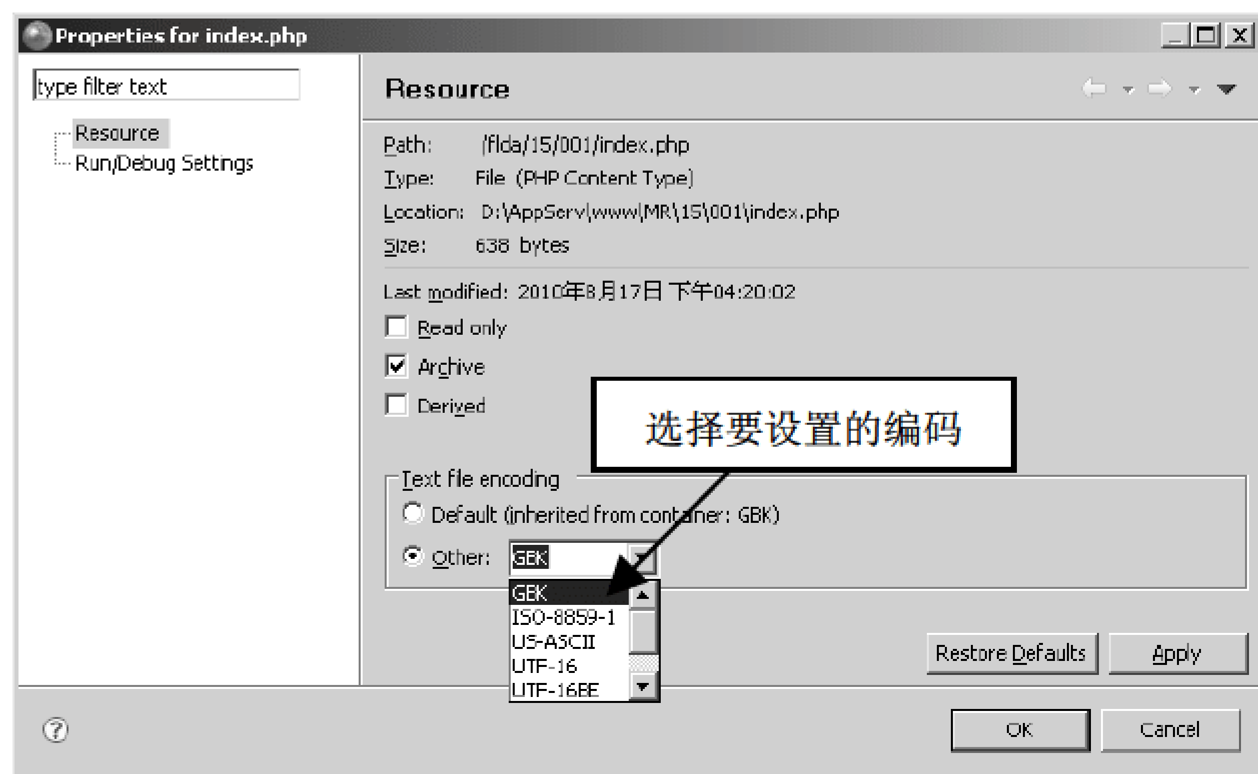


图 15.3　页面属性对话框

说明：在文件编码下拉列表中并没有 GB2312 编码，而 GBK 编码包含简体中文编码，所以这里选择 GBK 编码。

完成对页面文件的编码设置后，还需要使用 meta 标记设置页面编码，其设置方法如下：

```
<meta http-equiv="content-type" content="text/html; charset=gb2312" />
```

设计过程

（1）创建 index.php 页面，并设置页面文件的编码为 GB2312，其具体设置方法已经在关键技术中进行讲解，这里不再赘述。

（2）在 index.php 文件中通过 HTML 语言的 meta 标记设置网页编码为 GB2312 格式，具体实现代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>设计 GB2312 编码格式的网页</title>
        <meta http-equiv="content-type" content="text/html; charset=gb2312" />
        <link href="css/style.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
        <br/>
        <div style="width: 487px; text-align:center; height:221px; line-height:221px; font-size:16px; color:blue; background:url(img/bg.jpg)">
            这是一个 GB2312 编码格式的页面
        </div>
    </body>
</html>
```

秘笈心法

心法领悟 582：在 Dreamweaver 开发工具中设置文件的编码格式。

在 Dreamweaver 开发工具中设置文件的编码格式。首先，打开 Dreamweaver 开发工具，单击“编辑”按钮，选择“首选参数”命令，在弹出的“首选参数”对话框中通过“默认编码”下拉列表框设置创建文件的编码格式，如图 15.4 所示，设置完成后单击“确定”按钮。

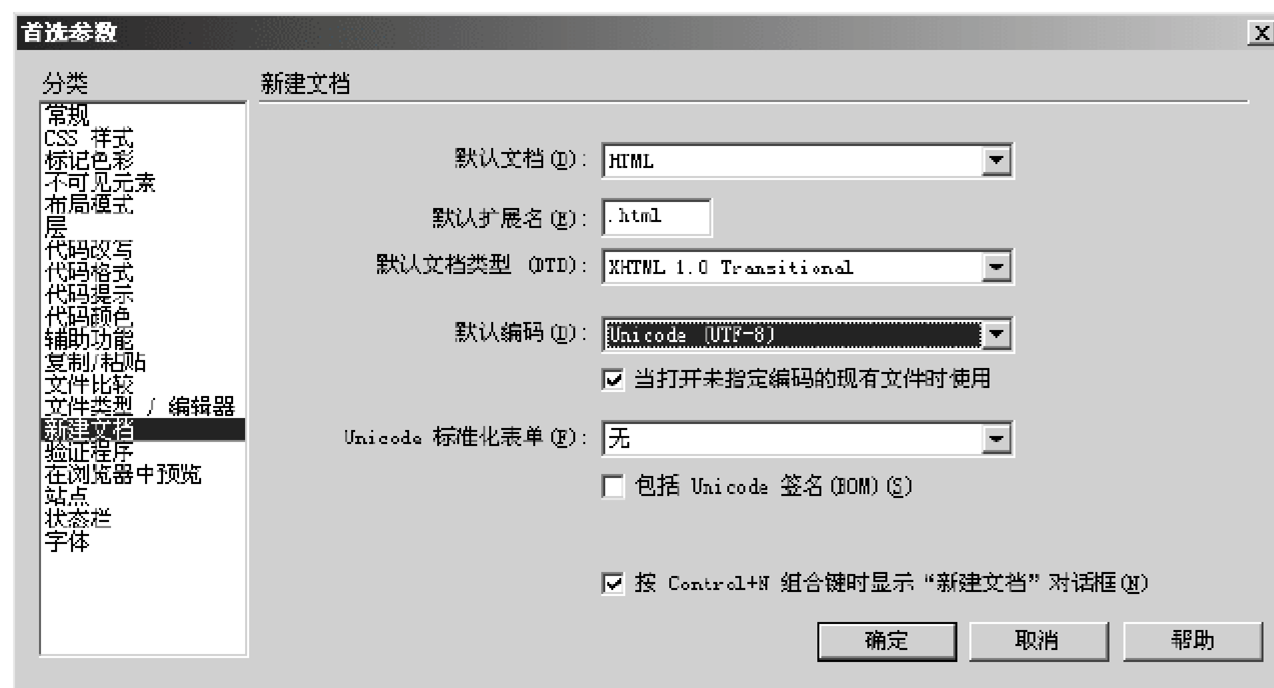


图 15.4 在 Dreamweaver 开发工具中设置文件的编码格式

实例 583 设计 GBK 编码格式的网页

光盘位置：光盘\MR\15\583

中级 趣味指数：★★★

实例说明

本实例主要讲解如何设置网页的页面编码格式为 GBK。运行本实例，如图 15.5 所示，在页面中打印出咖啡小屋程序所使用的页面编码。



图 15.5　打印咖啡小屋的页面编码

关键技术

本实例的关键技术是设置网页的页面编码格式为 GBK，同样其设置方法也包括两个部分。首先设置网页文件的编码为 GBK 格式，其设置方法与实例 582 相同，这里不再赘述。然后使用 HTML 语言的 meta 标记设置网页的编码为 GBK，其设定代码如下：

```
<meta http-equiv="content-type" content="text/html; charset=gbk" />
```

设计过程

（1）创建 index.php 文件，并使用 Zend Studio 编辑器设置页面文件编码格式为 GBK。

（2）在 index.php 文件中使用 meta 标记设置网页编码格式为 GBK，并在页面中打印当前页面所使用的编码，其具体代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>设计 GB2312 编码格式的网页</title>
        <meta http-equiv="content-type" content="text/html; charset=gbk" />
        <link href="css/style.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
        <br/>
        <div style="width: 498px; text-align:center; height:332px; line-height:221px; font-size:16px; color:#FF3300; background:url(img/bg.jpg)">
            这是一个 GBK 编码格式的页面
        </div>
    </body>
</html>
```

秘笈心法

心法领悟 583：GB2312 和 GBK。

GB2312 和 GBK 是中国政府颁布的标准汉字编码。GB2312 编码中放置了 6763 个常用汉字，并且首次在中文操作系统 CCDOS 中使用；GBK 编码在 GB2312 编码的基础上，增加了我国台湾 BIG5 编码中的 15000 多个汉字的字符。

实例 584	设计 UTF-8 编码格式的网页 光盘位置：光盘\MR\15\584	中级 趣味指数：★★★

实例说明

本实例主要讲解如何设置网页页面编码格式为 UTF-8。运行本实例，如图 15.6 所示，将在页面中打印出企

业邮局收发系统所使用的页面编码。



图 15.6　打印企业邮局收发系统页面编码

关键技术

UTF 是国际字符集转换格式（Unicode/UCS Transformation Format）的缩写，是 FCS 的实际表达式，按其基本长度所用位数分为 UTF-8/16/32 这 3 种形式，其中 UTF-8 保持字符数字一个字节，其他的用不定长编码到最多 6 个字节，支持到 31 位编码。UTF 中最常用的就是 UTF-8 编码格式，UTF-8 使用 8 位字节编码，通常使用 1 到 3 位字节来表示一个字符，字符的编码长度为变长。

UTF-8 字符集不仅包括欧美字符，也包括中国、日本和韩国的全角字符。因此，在网页中使用 UTF-8 字符集可以使用户在不安装中文编码的情况下浏览到大部分中文字符。

本实例设置页面为 UTF-8 编码，首先需要使用编辑工具设置页面文件编码格式为 UTF-8，然后使用 HTML 的 meta 标记设置网页编码格式为 UTF-8，其设置方法如下：

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

设计过程

（1）新建 index.php 文件，并使用 Zend Studio 编辑器设置页面文件编码格式为 UTF-8。

（2）在 HTML 语言的 meta 标记中设定当前页面所使用的编码格式为 UTF-8，其具体代码如下：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>设计 UTF-8 编码格式的网页</title>
        <meta http-equiv="content-type" content="text/html; charset=utf-8" />
        <link href="css/style.css" rel="stylesheet" type="text/css" />
    </head>
    <body>
        <br/>
        <div style="width: 978px; text-align:center; height:591px; line-height:591px; font-size:16px; color:#09600C; background:url(img/bg.jpg)">
            这是一个 UTF-8 编码格式的页面
        </div>
    </body>
</html>
```

秘笈心法

心法领悟 584：Web 程序开发中页面编码格式的选择。

为了便于程序的国际化及使程序支持更多语言版本的操作系统，在项目开发中建议将页面设置为 UTF-8 编码格式。

实例 585　PHP 的国际化

光盘位置：光盘\MR\15\585

中级

趣味指数：★★★

实例说明

信息全球化的高速发展，使企业的市场也在逐步扩大，并逐渐走向世界。企业向世界展示自己，必不可少地要应用到互联网，在应用互联网的过程中必须要提供多国语言的支持。只有这样才能让更多的人了解企业。所以在开发程序的过程中，必须要提供对多国语言的支持。本实例以一个简单的企业内部管理系统为例，介绍如何通过 PHP 语言开发提供多国语言支持的程序。在实例中，实现文章的添加和浏览功能，并且提供对中文和英文两种语言的支持。中文文章添加页面如图 15.7 所示，英文文章添加页面如图 15.8 所示。



图 15.7　中文文章添加页面



图 15.8　英文文章添加页面

关键技术

本实例中应用到的关键技术如下：

（1）网页编码的设置。

在网页中，字符编码通过<head>标签中<meta>标签的 charset 属性来设置。设置网页编码的代码如下：

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>设置网页编码格式</title>
</head>
<body>通过 meta 标签的 charset 属性设置字符编码格式</body>
</html
```

（2）获取客户端语言选项，根据语言选项的值调用不同的语言配置文件。

$_SERVER['HTTP_ACCEPT_LANGUAGE']为服务器变量，用于获取客户端浏览器的语言设置。

本实例中，获取语言选项和调用不同语言配置文件的代码如下：

```
<?php
session_start();
$lang=$_SERVER['HTTP_ACCEPT_LANGUAGE'];          //获取语言选项
$lang=explode(';',$lang);                        //以“;”对字符串进行分隔
$lang=$lang[0];                                  //为数组赋值
$lang=explode(',',$lang);                        //以“,”对字符串进行分隔
$lang=$lang[0];                                  //为数组赋值
switch($lang){                                   //根据获取到的语言选项的值，为变量赋值
 case "en-us":
        $lang="en";
```

```
		break;
	case "zh-cn":
		$lang="ch";
		break;
	default :
		$lang="en";
		break;
}
if(isset($_SESSION['lang'])){
	$lang=$_SESSION['lang'];			//为 SESSION 变量赋值
}
include "lang/lang_$lang.php";			//调用不同的语言配置文件
?>
```

设计过程

（1）创建 conn 文件夹，编写 conn.php 文件，完成与数据库的连接，其代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111") or die('连接失败:' . mysql_error());
mysql_select_db("db_database15",$conn) or die ('数据库选择失败:' . mysql_error());
mysql_query("set names gb2312");
?>
```

（2）创建 lang 文件夹，编写 lang_ch.php 和 lang_en.php 文件，分别定义中文语言配置和英文语言配置文件，代码请参考附送光盘中的内容。

（3）创建 config.php 文件，获取浏览器中语言选项的值，根据语言选项的值调用不同的语言配置文件。其完整代码可以参考关键技术中的内容。

（4）创建 setlang.php 文件，根据超链接中传递的参数，为 SESSION 变量赋值，并且调用不同的操作页面，其代码如下：

```
<?php
session_start();
switch($_GET['lang']){							//获取超链接传递的语言选项的值
	case "en":
		$_SESSION['lang']="en";					//根据超链接的值，为 SESSION 变量赋值
		break;
	case "ch":
		$_SESSION['lang']="ch";
		break;
	default :
		$_SESSION['lang']="en";
		break;
}
if($_GET['link']=="addnew"){					//根据超链接的值，跳转到不同操作页面
	header("Location:addnew.php");
}elseif ($_GET['link']=="show"){
	header("Location:show.php");
}else{
	header("Location:index.php");
}
?>
```

（5）创建 index.php 文件，根据选择的语言不同，输出不同的页面和内容。首先，通过 require_once()语句调用不同的包含文件，完成数据库连接和语言配置。然后，执行查询操作，从数据表中读取指定的数据信息。最后，在页面中输出数据，其关键代码如下：

```
<?php
require_once('conn/conn.php');										//连接数据库
require_once('config.php');										//调用语言配置文件
$query_rs="select * from tb_articles where langver='$rs_filter' order by id desc";	//定义查询语句
$rs=mysql_query($query_rs,$conn) or die(mysql_error());					//执行查询操作
$row_rs=mysql_fetch_assoc($rs);									//获取查询结果的记录数
?>
```

（6）创建 add_data.php 文件，添加 from 表单，用于提交数据，并完成对表单中数据处理的操作。首先，通过 require_once()语句调用不同的包含文件，完成数据库连接和语言配置。然后，定义方法 GetSQLValueString()

完成对数据格式的转换。接着，获取表单中提交的数据，并应用 sprintf()函数和 GetSQLValueString()方法完成对数据格式的转换，将数据添加到数据表中。最后，设计文章添加页面，创建 form 表单，完成数据的提交操作，其关键代码如下：

```
<?php
require_once(' conn /conn.php');                                              //连接数据库
require_once('config.php');                                                    //配置语言
//定义格式转换的方法
function GetSQLValueString($theValue,$theType,$theDefinedValue="",$theNotDefinedValue=""){
 $theValue=(!get_magic_quotes_gpc()) ? addslashes($theValue):$theValue;
 switch($theType){
        case "text";
                $theValue=($theValue !="")? "'". $theValue. "'" : "NULL";
                break;
 }
 return $theValue;
}
if($_POST["Submit"]==TRUE){
//获取表单中提交的数据，并完成对数据格式的转换
 $insertSQL=sprintf("insert into tb_articles(topic,author,comefrom,content,langver) values(%s,%s,%s,%s,%s)",
        GetSQLValueString($_POST['topic'],"text"),
        GetSQLValueString($_POST['author'],"text"),
        GetSQLValueString($_POST['comefrom'],"text"),
        GetSQLValueString($_POST['content'],"text"),
        GetSQLValueString($_POST['langver'],"text")
        );
//执行添加语句，将数据添加到数据表中
 $Result1=mysql_query($insertSQL,$conn) or die(mysql_error());
 echo "<script>alert('$add_data_Suc'); window.location.href='add_data.php';</script>";
}
?>
```

（7）创建 look_over.php 文件，根据超链接传递的 ID 值，执行查询操作，输出指定文章的详细信息，代码请参考光盘中内容。

秘笈心法

心法领悟 585：设置数据库中数据输出的编码格式。

设置数据库中数据输出的编码格式在 conn.php 文件中进行，通过 mysql_query()函数完成，本实例中设置的是 gb2312 编码格式。切记，这里设置的编码格式必须与数据库中字段的编码格式相同，否则在输出数据库中数据时将出现乱码。

如果没有在 conn.php 文件中进行数据库编码格式的设置，在输出数据库中的数据时，如果页面编码格式与数据库中字段设置的编码格式不同，那么在输出中文字符串时有可能出现乱码。

15.2　网站公共文件设计

在进行项目开发时，为提高代码的重用率及方便系统维护，经常需要将 Web 程序中的共性功能进行封装，本节将通过具体实例讲解网站公共文件的设计方法和具体应用。

实例 586	数据库连接类 光盘位置：光盘\MR\15\586	中级 趣味指数：★★★

实例说明

明日科技书店网站的数据库管理采用 ADODB 数据库抽象层技术，使用 ADODB 技术连接数据库，可以实



```
                                                            //通过 Connect()方法建立与数据库的连接
            if ($this->dbType == 'mysql') {
                $this->connID->Execute('set names gb2312');              //设置数据库编码
            }
        } elseif ($this->dbType == 'access') {                           //如果为 Access 数据库
            $this->connID->Connect('Driver = {Microsoft Access Driver (*.mdb)}; Dbq =' . realpath($this->dbName) . ';Uid = ' . $this->userName . ';Pwd
= ' . $this->password . ';');
        } else {
            return false;
        }
        $this->connID->debug = $this->isDebug;                           //是否调试
        return $this->connID;                                            //返回数据库连接对象
}
```

上述代码首先使用 require_once()函数包含 ADODB 中的 adodb.inc.php 文件，这样就实现了将 ADODB 类库包含到工程中，然后根据属性$dbType 的值判断要连接数据库的类型，根据不同的数据库类型选择不同的连接方式，最后通过 return 语句返回数据库连接对象。在进一步开发中，只需通过该方法返回的数据库连接对象来调用相应的方法（如 Execute()方法）即可实现对数据库的查询。

（4）完成对数据库的操作后，还需要关闭对数据库的连接，这里通过 closeConnID()方法实现，其代码如下：

```
function closeConnID (){
        @$this->connID->Disconnect();                                    //关闭与数据库的连接
}
```

以上为数据库连接类 ConnDB 的全部核心内容，主要实现对不同类型数据库的连接及在操作完数据库后释放资源等操作。

设计过程

（1）建立数据库连接类 ConnDB，其具体实现代码已经在关键技术中进行讲解，这里不再赘述。

（2）在 lzhConfig.ini 文件中配置各项数据库连接参数的值，具体配置代码如下：

```
[database-config]
dbType = mysql
host = 127.0.0.1
userName = root
password = 111
dbName = db_database15
isDebug = false
```

（3）对 ConnDB 类进行实例化，并使用实例化后的对象调用类中的 getConnID()方法返回数据库连接标识，从而实现与数据库的连接，代码如下：

```
$connDB = new ConnDB($arrayIni['dbType'] ,$arrayIni['host'] ,$arrayIni['userName'] ,$arrayIni['password'] ,$arrayIni['dbName'] ,$arrayIni['isDebug']);
$connID = $connDB->getConnID();
```

秘笈心法

心法领悟 586：屏蔽程序中的运行错误。

在开发 PHP 项目时，可以通过在语句前加“@”符号来屏蔽运行时的错误。

实例 587	数据库管理类 光盘位置：光盘\MR\15\587	中级 趣味指数：★★★

实例说明

为了对项目中的 SQL 语句进行统一管理，提高代码重用率和项目的可维护性，本实例将数据库的增、删、改、查操作通过专门的数据库管理类实现，在明日科技书店网站获取图书内容简介就是使用该类实现的，其运行效果如图 15.10 所示。



之所以要在数据库管理类中提取 SQL 语句的前 6 个字符，是利用 select、update 和 delete 关键字的字符长度都是 6 个字节的特点，通过截取的 6 个字节可以判断出 SQL 语句的类型。

实例 588　数据库分页类　中级

光盘位置：光盘\MR\15\588　趣味指数：★★★

实例说明

为了有效地利用页面资源，设计出合理的页面布局，在制作明日科技书店的图书信息展示模块时，采用分页的方式，明日科技书店的新书分页展示模块如图 15.11 所示。



图 15.11　图书分页展示

关键技术

如果每个图书信息展示模块都编写类似的代码实现分页，不仅会降低开发效率，还为生产环境下的维护带来很大负担，为解决上述问题，将图书的分页功能封装成分页类，只需在需要使用分页展示图书信息的页面调用该类的相应方法即可。下面介绍分页类的实现方法，其代码如下：

```
class PageDB{
    function pageData ($sql, $connID, $pageSize, $page){
        $arrayPageInfo = array();                                   //用于保存分页信息的数组
        $nowPage = $page;                                           //当前页码
      $rs = $connID->PageExecute($sql, $pageSize, $nowPage);        //通过 ADODB 的 PageExecute()方法获得指定页码的记录集
        $arrayData = $rs->GetRows();                                //将结果保存到数组中
        if (count($arrayData) == 0 || $rs == false) {               //如果没有查询到内容，则返回 FALSE
            return false;
        } else {
            $arrayPageInfo['data'] = $arrayData;                    //每页的数据
          $rsAll = $connID->Execute($sql);
            $countRs = count($rsAll->GetRows());
          $countPage = ceil($countRs / $pageSize);
            $arrayPageInfo['pageSize'] = $pageSize;                 //每页显示记录的条数
            $arrayPageInfo['countRs'] = $countRs;                   //总记录数
            $arrayPageInfo['page'] = $nowPage;                      //当前页
```

```
            $arrayPageInfo['countPage'] = $countPage;                    //总页数
            $arrayPageInfo['first'] = 1;                                 //第一页的页码
            if ($page > 1) {
                $arrayPageInfo['previous'] = $rs->AbsolutePage() - 1;    //前一页的页码
            } else {
                $arrayPageInfo['previous'] = 1;
            }
            if ($page < $countPage) {
                $arrayPageInfo['next'] = $rs->AbsolutePage() + 1;        //后一页的页码
            } else {
                $arrayPageInfo['next'] = $countPage;
            }
            $arrayPageInfo['last'] = $countPage;                         //最后一页的页码
        }
        return $arrayPageInfo;
    }
}
```

上述分页类实现的原理是：设法使用 ADODB 数据抽象层获取指定页码中所显示的数据，同时将首页、上一页、下一页、尾页、总页码和总记录数都作为数组元素，然后将该数组作为分页类中实现分页方法的返回值返回，在需要分页的模块，只需要通过分页类获取该返回数组就可以获得所有分页所需信息。

在实现上述 PageDB 分页类时，在类体内只定义一个 pageData()分页方法，在该方法内部，通过 ADODB 数据抽象层根据分页方法的$pageSize 和$page 参数获取页面显示的数据和分页导航参数，并保存到$arrayPageInfo 数组中返回。

设计过程

（1）建立数据库分页类 PageDB，并在该类中建立 pageData()方法，该类的创建过程已经在关键技术中给出，这里不再赘述。

（2）在 listsepbook.php 文件中，包含网站头文件，根据超链接传递的参数值为变量赋值，调用数据库管理类中的方法执行查询语句，并且将返回的查询结果赋给模板变量。定义分页变量，调用分页类中的方法完成数据的分页输出，同时将分页方法返回的数据赋给模板变量，最后指定模板页，其关键代码如下：

```
require_once 'header.php';                                                  //包含 header.php 文件
if ($_GET['t'] == 'new') {                                                  //展示新书
    $nowtype = '新书推荐';
    $andwhere = "isnew = 1";
} elseif ($_GET['t'] == 'sepprice') {                                       //展示特价图书
    $nowtype = '特价图书';
    $andwhere = "issepprice = 1";
} elseif ($_GET['t'] == 'hotsell') {
    $nowtype = '热卖图书';                                                  //展示热卖图书
    $andwhere = "ishotsell = 1";
} elseif ($_GET['t'] == 'term') {
    $nowtype = '期书';
    $andwhere = "isterm = 1";
} elseif ($_GET['t'] == 'mrbook') {                                         //展示明日图书
    $nowtype = '明日图书推荐';
    $andwhere = "ismrbooktj = 1";
}
$smarty->assign('gt', $_GET['t']);                                          //为模板变量赋值
$smarty->assign('nowtype', $nowtype);
$system = $adminDB->executeSQL("select bookimgurl, readurl from tb_system where mark = 1", $connID);    //执行查询语句
$smarty->assign('system', $system);                                         //将查询结果赋给模板变量
if (! isset($_GET['page']) || $_GET['page'] == '') {                        //判断分页变量的值
    $page = '1';                                                            //设置分页变量的值
} else {
    $page = $_GET['page'];
}
$sql = "select tb_bookinfo.id, tb_bookinfo.bookimg, tb_bookinfo.about, tb_bookinfo.browsertime, tb_bookinfo.oldprice, tb_bookinfo.newprice,
tb_bookinfo.bookname, tb_bookinfo.writer, tb_bookinfo.pubtime, tb_bookinfo.addtime, tb_bookinfo.ishave, tb_pub.pubname from tb_bookinfo, tb_pub
where tb_bookinfo.pubid = tb_pub.id and " . $andwhere . " order by tb_bookinfo.addtime desc";             //定义查询语句
$bookinfos = $pageDB->pageData($sql, $connID, 10, $page);                   //调用分页类中的方法，执行查询操作
```

```
$smarty->assign('bookinfos', $bookinfos);                //将分页方法返回的数组赋给模板变量
$smarty->display('listsepbook.phtml');                   //指定模板页
require_once 'footer.php';                               //包含页面的尾文件
```

（3）在 views\ listsepbook.phtml 模板页中完成数据的分页输出，其代码请参考光盘中的内容。

秘笈心法

心法领悟 588：分页方法的完善。

在本实例中，采用的是首页、上一页、下一页和尾页的分页模式，如果读者感兴趣也可以对这种分页模式进行更新，使其更加个性化，分页类存储于 library\ PageDB.php 文件中。

实例 589　Smarty 模板引擎配置类

光盘位置：光盘\MR\15\589

中级　趣味指数：★★★

实例说明

Smarty 是 PHP 的模板引擎，通过 Smarty 可以将开发人员和美工人员相互独立开来，在项目开发中，后台编码和前台设计可以同步进行。在对项目进行维护的过程中，页面更改不会影响业务逻辑的实现。同时，Smarty 模板增加了视图缓存和静态页面生成机制，从而可以有效提高项目的执行效率。

明日科技图书网站就是应用 Smarty 模板完成网站的动静分离的，其动态文件存储于项目的根目录下，模板文件存储于根目录下的 views 文件夹中，而 Smarty 的缓存文件和编译文件则存储于 library\smarty 目录下，如图 15.12 所示。

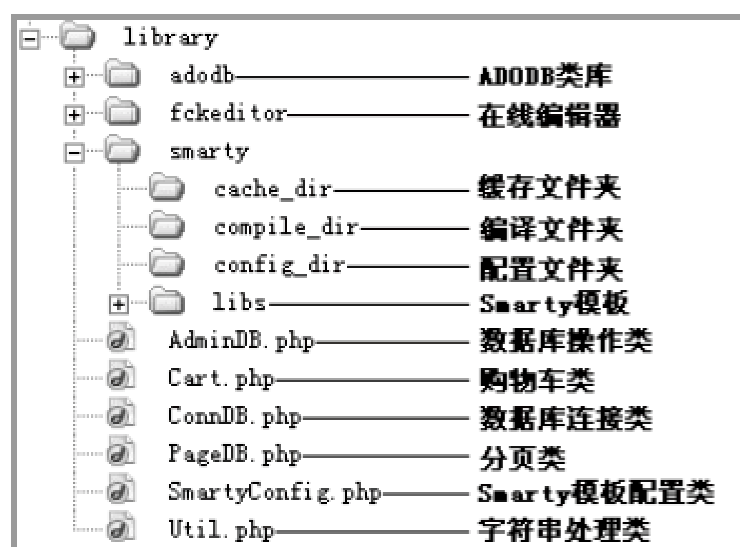


图 15.12　明日科技图书网的类文件存储位置

关键技术

Smarty 模板引擎的配置类封装在 SmartyConfig.php 文件中，其代码如下：

```
require_once 'library/smarty/libs/Smarty.class.php';     //包含 Smarty.class.php 文件，从而导入 Smarty 类库
class SmartyConfig extends Smarty {                      //编写 SmartyConfig 类，使其继承自 Smarty 类
    function SmartyConfig (){                            //构造方法
        $this->Smarty();                                 //调用父类的构造方法
        $arrayIni = parse_ini_file('config/lzhConfig.ini');  //解析 LzhConfig.ini 文件，来获取配置信息
        $this->template_dir = $arrayIni['template_dir'];     //模板文件保存目录
        $this->compile_dir = $arrayIni['compile_dir'];       //编译文件保存目录
        $this->cache_dir = $arrayIni['cache_dir'];           //缓存文件保存目录
        $this->config_dir = $arrayIni['config_dir'];         //配置文件保存目录
        $this->caching = $arrayIni['caching'];               //是否开启视图缓存
    }
}
```

技巧：函数 parse_ini_file()用于解析.ini 文件，并将解析结果存储到数组中。



表 15.1　Util 类中的方法列表

方　　法	说　　明
Util ()	构造方法，载入配置文件
baseUrl ()	定义网站中地址的相对路径
moneyFormat($money)	对货币进行格式化。参数$money 为金额
unHtml ($text)	对 html 标记进行转义输出，参数$text 为转义的文本
msubstr ($str, $start, $len)	对中文进行截取，防止出现乱码。参数$str：要截取的字符串；$start：开始截取的位置；$len：截取的长度
editor ($name, $value, $width = '100%', $height = '200')	集成 FCKEditor 编辑器。参数$name：编辑器名称；$value：编辑器内容；$width：编辑器宽度；$height：编辑器高度
editorUnHtml ($text)	转义编辑器输出内容。参数$text 指输出内容
setRed ($arrayKey, $text)	对关键字描红。参数$arrayKey：查询的对象；$text：查询的关键字

设计过程

在 library 文件夹下新建 Util.php 文件，封装字符串类 Util，定义表 15.1 中列出的字符串操作方法。

（1）首先定义该类的属性，充分考虑到项目在生产环境下的维护工作，将类的常用配置都写入到 LzhConfig.ini 文件，所以该类的属性只需要一个用来保存配置文件信息的数组即可，代码如下：

```
var $arrayIni;                                          //用于保存 LzhConfig.ini 配置文件的配置信息
```

（2）定义工具类的构造方法。在构造方法中通过函数 parse_ini_file()解析 LzhConfig.ini 配置文件，实现代码如下：

```
function Util (){
    $this->arrayIni = parse_ini_file('config/lzhConfig.ini');          //解析配置文件，并将结果保存到数组中
}
```

（3）一般情况下，项目在开发环境和生产环境下的根目录是不同的，如果开发过程中采用相对路径，改动较小，如果使用绝对路径，那么所有涉及路径的模块在正式发布项目时都需要改动。为了解决上述问题，可以将项目根目录在 LzhConfig.ini 文件中指定，然后在需要使用项目路径的模块中读取配置文件所指定的项目根目录即可，这里获取项目根目录应用的是 baseUrl()方法，代码如下：

```
function baseUrl (){
    $arrayIni = $this->arrayIni;
    return $arrayIni['baseUrl'];                                       //将配置文件中的网站根目录返回
}
```

（4）作为电子商务类网站，项目的很多模块中都需要实现对货币金额的格式化输出，为了简化操作，在字符串类中定义 moneyFormat()方法，其代码如下：

```
function moneyFormat ($money){
    return str_replace(',', '', number_format($money, 2));             //返回中文货币格式
}
```

利用 moneyFormat()方法首先通过函数 number_format()将金额保留 2 位小数后格式化输出，然后将格式化后高位的“,”替换为空，最后将处理结果返回。

（5）为了保持所录入数据按原格式输出，需要将一些特殊字符转义成 HTML 标记，由于在项目的很多模块需要使用到该功能，所以将该功能作为一个方法封装到字符串处理类中，其代码如下：

```
function unHtml ($text){
    $str = htmlspecialchars($text);                                    //将特殊字符替换成 HTML 转义标记
    $str=str_replace(chr(13),"<br>",$str);                             //将换行替换成标记<br/>
    $str = nl2br($str);                                                //在每个新行前加入 HTML 换行标记<br/>
    $str=str_replace(chr(32)," ",$str);                           //将空格替换成标记$nbsp;
    return $str;                                                       //返回处理后字符串
}
```

（6）使用 PHP 提供的函数 substr()截取中文字符会导致截取乱码，而且在制作项目时，很多模块都需要对过长字符进行截取，所以在字符串处理类中定义 msubstr()方法，对中文字符进行截取，其代码如下：

```
function msubstr ($str, $start, $len){
    $strlen = $start + $len;                                    //获取截取字符长度
    for ($i = 0; $i < $strlen; $i ++) {                         //遍历所有字符
        if (ord(substr($str, $i, 1)) > 0xa0) {                  //判断是否为中文字符
            $tmpstr .= substr($str, $i, 2);                     //如果是中文则截取 2 个字节
            $i ++;
        } else
            $tmpstr .= substr($str, $i, 1);                     //不是中文则截取 1 个字节
    }
    return $tmpstr;                                             //返回结果
}
```

（7）使用 HTML 语言提供的文件域对文本进行编辑。实现文本特效较复杂，而且不能达到“所见即所得”的效果，为了解决上述问题，在进行 Web 项目开发时候，一般使用 FckEditor 在线编辑器作为表单的文本编辑域。在制作明日科技书店时会在很多模块使用到在线编辑器，所以在开发时将调用 FckEditor 编辑器的代码封装在 Util 字符串处理类中，具体代码如下：

```
function editor ($name, $value, $width = '100%', $height = '200'){
    require_once 'library/fckeditor/fckeditor.php';                  //导入 FckEditor 类库
    $arrayIni = $this->arrayIni;                                       //获取配置文件内容
    $oFCKeditor = new FCKeditor($name);                                //实例 FCKeditor
    $oFCKeditor->BasePath = $arrayIni['baseUrl'] . '/library/fckeditor/';  //设置 FCKeditor 路径
    $oFCKeditor->Width = $width;                                       //编辑器的宽度
    $oFCKeditor->Height = $height;                                     //编辑器的高度
    $oFCKeditor->ToolbarSet = 'Default';                               //设置工具栏样式
    $oFCKeditor->Value = $value;                                       //初始值
    $oFCKeditor->Create();                                             //创建编辑器
}
```

（8）完成 Util 类的封装，在 lzh.inc.php 文件中，应用 Smarty 中的 register_object()方法将其注册为模板对象，在模板页中通过模板对象调用类中方法实现对字符串的操作。

在 lzh.inc.php 文件中不只实现注册模板对象的功能，所有类的实例化操作都在该文件中完成，并返回实例化对象，包括数据库连接类、数据库操作类、分页类和 Smarty 配置类，其代码如下：

```
<?php
session_start();                                                //初始化 SESSION 变量
$arrayIni = parse_ini_file('config/lzhConfig.ini');             //包含配置文件
require_once 'library/ConnDB.php';                              //载入数据库连接类文件
require_once 'library/AdminDB.php';                             //载入数据库操作类文件
require_once 'library/PageDB.php';                              //载入分页类文件
require_once 'library/SmartyConfig.php';                        //载入 Smarty 配置类文件
require_once 'library/Cart.php';                                //载入购物车类文件
require_once 'library/Util.php';                                //载入字符串处理类文件
$connDB = new ConnDB($arrayIni['dbType'] ,$arrayIni['host'] ,$arrayIni['userName'] ,$arrayIni['password'] ,$arrayIni['dbName'] ,$arrayIni['isDebug']);
$connID = $connDB->getConnID();                                 //执行与数据库的连接
$adminDB = new AdminDB();                                       //数据库操作类实例化
$pageDB = new PageDB();                                         //分页类实例化
$smarty = new SmartyConfig();                                   //Smarty 配置类实例化
$util =   new Util();                                           //字符串操作类实例化
$smarty->register_object('util', $util, null, false);          //注册模板对象
```

（9）在 index.php 文件中，调用数据库操作类中的方法，查询出数据库中的数据，并且将查询结果赋给模板变量，最终指定模板页。

（10）在模板页 views\index.phtml 中，应用 section 语句循环输出模板变量中的数据，并且通过注册对象调用字符串类中的方法对输出的图书标题、图书价格和公告标题等进行格式化操作，其关键代码如下：

```
{section name=rID loop=$reads}
<div style="width:198px; padding-bottom:7px">
 <div style="width:100%; height:90px">
        <div style="width:85px; height:100px; float:left">
                <a href="{util->baseUrl}/bookinfo-{$reads[rID].bookid}.html"><img src="{$system[0].bookimgurl}/{$reads[rID].bookimg}"
width="70" height="90"    style="border:1px solid #123456" /></a>
          </div>
          <div style="width:112px; height:100px; float:right; text-align:left">
                <li style="display:inline; height:35px"><a href="{util->baseUrl}/bookinfo-{$reads[rID].bookid}.html" class="a6">•{util->unHtml
p=$reads[rID].bookname}</a></li>
```





包含文件是指将另一个文件中的内容包含到当前文件中来。包含外部文件可以提高代码的重用率，是 PHP 编程的重要技巧。PHP 提供了 include、require、include_once 和 require_once 语句用于实现包含文件操作。

使用 require 语句调用文件时，如果调用的文件没找到，require 语句会输出错误信息，并且立即终止脚本的处理。而 include 语句在没有找到文件时会输出警告，但不会终止脚本的处理。

使用 require 语句调用外部文件时，只要程序一执行，就会立刻调用；而使用 include 语句时，只有程序执行到该语句时，才会调用外部文件。

实例 592	网站的尾文件设计 光盘位置：光盘\MR\15\592	中级 趣味指数：★★★

实例说明

明日科技图书网尾文件也是作为单独的文件进行存储的，其运行效果如图 15.15 所示。



图 15.15　网站尾文件的运行效果

关键技术

网站尾文件在输出各种超链接的同时，还输出数据库中 tb_system 表中的数据，其存储的是项目的系统信息。tb_system 表的结果如图 15.16 所示。

在运行明日科技图书网的程序时，要对 tb_system 表中 readurl、bookimgurl 和 ggurl 3 个字段的值进行修改，以设置正确的程序路径，如图 15.17 所示。

服务器: localhost ▸ 数据库: db_database15 ▸ 表 : tb_system

字段	类型	整理	属性	Null	默认	额外
id	int(4)			否		auto_increment
merid	varchar(50)	gb2312_chinese_ci		否		
meracct	varchar(50)	gb2312_chinese_ci		否		
mark	tinyint(1)			否	0	
readurl	varchar(100)	gb2312_chinese_ci		否		
bookimgurl	varchar(200)	gb2312_chinese_ci		否		
ggurl	varchar(200)	gb2312_chinese_ci		否		
bq	varchar(200)	gb2312_chinese_ci		否		
address	varchar(250)	gb2312_chinese_ci		否		
tel	varchar(50)	gb2312_chinese_ci		否		
cz	varchar(50)	gb2312_chinese_ci		否		
email	varchar(200)	gb2312_chinese_ci		否		
icp	varchar(50)	gb2312_chinese_ci		否		
qq	varchar(50)	gb2312_chinese_ci		否		

图 15.16　明日科技图书网的系统表结果

readurl	bookimgurl	ggurl
http://127.0.0.1/mr/15/005/upfiles/doc	http://127.0.0.1/mr/15/005/upfiles/bookimg	http://127.0.0.1/mr/15/005/upfiles/ggimg

图 15.17　设置程序在服务器中的相对路径

笔者在开发本项目时，将其存储于 Apache 服务器的 mr/15/005 文件夹下。读者在运行程序时，需要根据自己的实际情况对这个路径进行修改。

设计过程

（1）新建 footer.php 文件，读取项目系统表 tb_system 中的数据，并且将数据存储到模板变量中，同时指定模板页，其代码如下：

```
<?php
```



（2）将 lzhScrollImg.js 脚本文件存储于本实例根目录的 js 文件夹下。

秘笈心法

心法领悟 593：js 特效。

读者不必自己研究 js 脚本特效的实现方法，可以充分利用网络资源，查找自己喜欢的特效代码，然后将其应用到自己的程序中即可。

15.3　注册、登录

明日科技图书网用户登录模块的主要功能是为用户提供服务、建立网站管理者和用户沟通的桥梁，主要包括用户注册、用户登录、用户注册信息修改、用户密码修改、查看提交订单信息和查看在网站上的消费情况等功能。

实例 594	用户注册 光盘位置：光盘\MR\15\594	中级 趣味指数：★★★

实例说明

用户注册模块主要用于对用户注册信息进行收集，用户再次参与该站点活动时，通过登录模块成功登录后即可参与进来。用户注册模块不仅可以方便用户参与本站活动，而且也是对网站人气的积累。明日网上书店的用户注册表单如图 15.19 所示。

图 15.19　用户注册表单

关键技术

（1）图片验证码。

制作图片验证码时，首先通过 JavaScript 的 Math 对象的 random()方法产生一个随机数，并通过 Math 对象的 round()方法截取该随机数的前 4 位。然后再分别提取该随机数的每一位，并将提取的每一位数字通过 GET 方法传递到 xym.php 页面，在 xym.php 页面中根据传递过来的数组读取相应的图片，并将结果显示出来，其代

码如下：

```
$num=$_GET[num];                                        //获得传递的数字
$address="img/code/".$num.".gif";                       //获取对应图片地址
$fp=fopen($address,"r");                                //打开对应的图片
echo fread($fp,filesize($address));                     //将图片以二进制的形式输出到浏览器
fclose($fp);                                            //关闭打开的图片
```

在将生成的随机数的每一位传递到 xym.php 文件的同时，将该随机数赋给一个隐藏域，这样在验证用户输入的验证码是否正确时，就可以根据用户输入的内容和该隐藏域的 value 值进行对比得出结果，如果二者相等则说明用户输入的验证码正确。

（2）验证用户输入的用户名是否被占用。

当用户输入注册信息时，应用 JavaScript 对用户输入的内容进行实时验证，其中验证用户名是否被占用的方法存储于\js\fun.js 文件中，其代码如下：

```
function chkreginfo(form, mark, edit, baseUrl) {
    if ((mark == 0 || mark == "all") && edit != "edit") {
        if (form.usernc.value == "") {                      //判断用户输入的昵称是否为空
            chkusernc.innerHTML = "请输入用户昵称！ ";       //如果为空则给出提示
            form.usernc.style.backgroundColor = "#FF0000";  //更改用户昵称文本框的背景为红色
            return false;
        } else {
            var xmlobj;                                     //定义 XMLHttp 对象
            if (window.ActiveXObject) {                     //如果是 Windows 系统
                xmlobj = new ActiveXObject("Microsoft.XMLHTTP");    //通过 ActiveXObject()实例 XMLHttp 对象
            } else if (window.XMLHttpRequest) {
                xmlobj = new XMLHttpRequest();              //通过 XMLHttpRequest()实例 XMLHttp 对象
            }
            xmlobj.onreadystatechange = fun;                //设置回调函数
            xmlobj.open("GET",                              //通过 GET 方法向服务器发送数据
                    baseUrl + "/chkunc.php?unc=" + form.usernc.value, true);
            xmlobj.send(null);
            function fun() {                                //定义回调函数
                if (xmlobj.readyState == 4 && xmlobj.status == 200) {
                    if (xmlobj.responseText == "0") {       //如果返回 0 则说明昵称被占用
                        chkusernc.innerHTML = "该昵称已被占用！ ";
                        form.usernc.style.backgroundColor = "#FF0000";
                        return false;
                    } else {
                        chkusernc.innerHTML = "";
                        form.usernc.style.backgroundColor = "#FFFFFF";
                    }
                }
            }
        }
    }
......    //省略代码请详见本书附带光盘
    return true;
}
```

上述代码首先根据不同浏览器的类型生成 HTTP 请求对象，然后使用该对象的 open()方法将用户输入的昵称传递给 chkunc.php 页面，并根据 chkunc.php 页面的返回值判断用户输入的昵称是否已经被占用，其中 chkunc.php 的关键代码如下：

```
require_once 'lzh.inc.php';                             //包含公共类
$arrayunc=$adminDB->executeSQL("select id,usernc from tb_user where usernc='".$_GET["unc"]."'",$connID);    //查找用户昵称
if($arrayunc)
    echo 0;                                             //如果查找到则返回 0
else
    echo 1;                                             //没有查到则返回 1
```

设计过程

（1）制作用户注册表单 UI 界面。用户注册表单除用户名和登录密码等信息外，为了与用户进行沟通，还需要用户的联系电话、E-mail 地址和邮编等信息，据此编写明日科技书店的注册表单，模板页 getuserinfo.phtml

的主要代码如下：

```
<form name="form_reg" method="post" action="{util->baseUrl}/getuserinfo.php" onSubmit="return chkreginfo(form_reg,'all')">
<table width="90%" border="0" align="center" cellpadding="0" cellspacing="0">
 <tr>
      <td width="22%" height="30"><div align="right">用户昵称：</div></td>
      <td colspan="3"> <input type="text" name="usernc" size="20" class="input" onBlur="chkreginfo(form_reg,0, '', '{util->baseUrl}')"></td>
      <td width="25%"><div id="chkusernc" style="color:#FF0000"></div></td>
    </tr>
    <tr>
      <td height="30"><div align="right">登录密码：</div></td>
      <td height="30" colspan="3"> <input type="password" name="userpwd1" size="25" class="input" onBlur="chkreginfo(form_reg,1)"
></td>
      <td height="30"><div id="chkuserpwd1" style="color:#FF0000"></div></td>
    </tr>
    ......<!—此处省略部分表单元素实现代码，请详见本书附带光盘-->
    <tr>
      <td height="30"><div align="right">验证码：</div></td>
      <td width="17%" height="0"> <input type="text" name="xym" size="8" class="input" onBlur="chkreginfo(form_reg,12)"/><input
type="hidden" value="" name="xym1"></td>
      <td width="12%">
      <script language="javascript">
          var num1=Math.round(Math.random()*10000000);
          var num=num1.toString().substr(0,4);
          document.write("<img name=codeimg4 src='{util->baseUrl}/xym.php?num="+num.substr(0,1)+"&baseUrl={util->baseUrl}'>");
          document.write("<img name=codeimg5 src='{util->baseUrl}/xym.php?num="+num.substr(1,1)+"&baseUrl={util->baseUrl}'>");
          document.write("<img name=codeimg6 src='{util->baseUrl}/xym.php?num="+num.substr(2,1)+"&baseUrl={util->baseUrl}'>");
          document.write("<img name=codeimg7 src='{util->baseUrl}/xym.php?num="+num.substr(3,1)+"&baseUrl={util->baseUrl}'>");
          form_reg.xym1.value=num;
<!—该方法用于实现重新为验证码赋值-->
          function code_1(){ldelim}
              var num1=Math.round(Math.random()*10000000);
              var num=num1.toString().substr(0,4);
              document.codeimg4.src="{util->baseUrl}/xym.php?num="+num.substr(0,1)+"&baseUrl={util->baseUrl}";
              document.codeimg5.src="{util->baseUrl}/xym.php?num="+num.substr(1,1)+"&baseUrl={util->baseUrl}";
              document.codeimg6.src="{util->baseUrl}/xym.php?num="+num.substr(2,1)+"&baseUrl={util->baseUrl}";
              document.codeimg7.src="{util->baseUrl}/xym.php?num="+num.substr(3,1)+"&baseUrl={util->baseUrl}";
              form_reg.xym1.value=num;
          {rdelim}
      </script></td>
      <td width="24%"><a href="javascript:code_1()" class="a1">看不清</a></td>
      <td height="0"><div id="chkxym" style="color:#FF0000"></div></td>
    </tr>
    <tr>
      <td height="40" colspan="5"><div align="center">
          <input type="submit" value="立即注册" class="btn_grey">
          <input type="reset" value="重置" class="btn_grey">
          </div>
       </td>
    </tr>
 </table>
</form>
```

（2）当用户输入注册信息时，应用 JavaScript 对用户输入的内容进行实时验证，其 JavaScript 脚本存储于\js\fun.js 文件中，完整代码请参考本书光盘。

（3）用户输入的内容成功通过数据有效性验证后，所有输入的内容将被提交到 getuserinfo.php 页面进行保存，其关键代码如下：

```
require_once 'header.php';                                        //包含 header.php 文件
if (isset($_POST["usernc"]) && $_POST["usernc"] != "") {           //判断是否提交了表单
    if (! $adminDB->executeSQL("select id, usernc from tb_user where usernc='" . $_POST["usernc"] . "'", $connID)) {    //判断昵称是否被占用
        $time = date("Y-m-d H:i:s");                               //获取注册时间
        if ($adminDB->executeSQL("insert into
tb_user(usernc,truename,pwd,email,sex,tel,qq,address,logintimes,regtime,lastlogintime,ip,yb,usertype,answer,question,truepwd,photo,score) values('" .
$_POST["usernc"] . "','" . $_POST["truename"] . "','" . md5(trim($_POST["userpwd1"])) . "','" . $_POST["email"] . "','" . $_POST["sex"] . "','" .
$_POST["tel"] . "','" . $_POST["qq"] . "','" . $_POST["address"] . "','0','$time','$time','" . $_SERVER['REMOTE_ADDR'] . "','" . $_POST["yb"] . "','0','" .
$_POST["answer"] . "','" . $_POST["question"] . "','" . trim($_POST["userpwd1"]) . "','" . "img/face/" . $_POST["photo"] . "','30')", $connID)) {
```

```
//保存用户的注册信息
            if (isset($_SESSION["unc"]) && $_SESSION["unc"] != "") {
                unset($_SESSION["unc"]);
            }                                       //销毁原来登录用户的登录昵称
            session_register("unc");                //注册新的用户昵称到 SESSION 变量
            $_SESSION["unc"] = trim($_POST["usernc"]);
          if (isset($_SESSION['toUrl'])) {          //获取要跳转的页面，并进行跳转
                echo "<script>window.location.href='" . $util->baseUrl() . "/" . $_SESSION['toUrl'] . "';</script>";
            } else {
                echo "<script>window.location.href='" . $util->baseUrl() . "/regsuccess.html';</script>";
            }
        } else {
            echo "<script>alert('注册失败，请重试！')</script>";
        }
    } else {
        echo "<script>alert('该昵称已经被他人注册！')</script>";
    }
}
$smarty->display('getuserinfo.phtml');
require_once 'footer.php';
```

秘笈心法

心法领悟 594：防止用户越过表单限制上传数据。

在本项目的 getuserinfo.php 文件中，为了避免由于用户的浏览器禁用 JavaScript 脚本功能而使用户输入昵称的唯一性验证失效，或者某些用户越过表单的验证机制直接上传数据，在保存用户提交的注册信息前，首先使用数据库管理类的 executeSQL()方法查询是否存在和用户提交昵称相同的注册用户，如果存在则提示该昵称已被占用，并返回注册页面要求用户重新填写注册信息，反之则将用户提交的注册信息保存到数据库中。

实例 595 用户登录

光盘位置：光盘\MR\15\595

高级

趣味指数：★★★☆

实例说明

用户在明日科技图书网购买图书时，需要首先登录站点，这样可以记录购买人信息，并且可在用户中心的订单查询模块中查询到该用户的所有订单信息。明日科技图书网的用户登录页面如图 15.20 所示。

图 15.20　用户登录页面

关键技术

登录的目的就是为用户开据一张通行证，使其可以在网站中进行任意操作（权限范围之内），也是其行使网站会员权限的依据。而这个通行证的开据应用的是 SESSION 技术。登录成功后，将用户名存储到 SESSION 变量中，即完成通行证的开据，其代码如下：

```
$_SESSION['unc'] = $usernc;
```

注意： *在应用 SESSION 变量之前，切记一定要应用 session_start()函数初始化 SESSION 变量，否则 SESSION 变量不能使用，并且在应用 session_start()函数之前不能有任何输出。本项目中 SESSION 变量的初始化操作是在包含文件 lzh.inc.php 中完成的。*

该功能利用 SESSION 可以在页面之间传递数据的特性，确保这个通行证在网站的任意一个页面中都有效，直到用户退出登录。

设计过程

（1）在 views 文件夹下创建 login.phtml 登录表单的 UI 界面，其实现方法与注册表单类似，这里不再赘述。

（2）当用户单击“登录”按钮后，将对用户输入的用户名、密码和验证码进行有效性验证，如果验证通过则将用户输入的数据提交到 login.php 中进行处理，关键代码如下：

```
<?php
require_once 'header.php';
$errorMsg = '';
if (isset($_POST['usernc']) && $_POST['usernc'] != '') {              //判断是否提交了表单
    $usernc = trim($_POST['usernc']);                                 //获得提交的用户名
    $pwd = trim($_POST['userpwd']);                                   //获得提交的登录密码
    $user = $adminDB->executeSQL("select id, usernc from tb_user where usernc='" . $usernc . "' and pwd='" . md5($pwd) . "'", $connID); //按用户名和密码查询用户
    if (! $user) {
        $errorMsg = '用户名或密码输入有误！ ';
    } else {
        if (isset($_SESSION["unc"])) {                                //销毁原登录用户信息，并注册新登录用户信息给 SESSION 变量
            session_unregister('unc');
        }
        $_SESSION['unc'] = $usernc;
        @$adminDB->executeSQL("update tb_user set logintimes=logintimes+1 ,lastlogintime ='" . date('Y-m-d H:i:s') . "',ip='" .
$_SERVER['REMOTE_ADDR'] . "' where usernc='" . $_SESSION['unc'] . "'", $connID);        //更改用户的登录信息
        if (isset($_SESSION['toUrl'])) {
            echo "<script>window.location.href='" . $util->baseUrl() . "/" . $_SESSION['toUrl'] .."';</script>";
            exit();
        } else {
            echo "<script>window.location.href='" . $util->baseUrl() . "/usercenter.html';</script>";
            exit();
        }
    }
}
$smarty->assign('errorMsg', $errorMsg);
$smarty->display('login.phtml');
require_once 'footer.php';
```

上述代码中，首先定义变量$errorMsg，保存处理登录表单提交数据时出现的错误信息（例如登录失败信息等），然后使用$_POST 全局数组接收用户提交的用户名和登录密码，并使用数据库管理类的 executeSQL()方法执行查询语句来判断用户输入的用户名和密码是否正确，如果正确则更改用户登录次数和最后登录时间，反之则将登录失败信息赋给变量$errorMsg，并传递给 login.phtml 模板。

秘笈心法

心法领悟 595：延长 SESSION 的生命周期。

在开发程序的过程中可能会遇到这样的情况，以会员的身份登录一个网站后，若停留的时间较长，此时在这个网站中发布信息时，会提示“您没有登录，请登录后再发布信息”。

之所以出现此类问题，就是因为在登录时定义的 SESSION 变量已经过期，导致当前用户不可用，需要重新登录。解决这个问题最好的方法就是通过 session_set_cookie_params()函数延长 SESSION 的生命周期，将过期时间设置为 0。

实例 596	用户中心 光盘位置：光盘\MR\15\596	高级 趣味指数：★★★★☆

实例说明

明日科技图书网的用户中心，主要用于实现对用户注册信息和用户登录密码的修改、用户订单的查询和意



本框、新密码录入文本框和确认密码录入文本框 3 个表单元素，如图 15.22 所示。

系统将用户修改的数据提交到 usercenterchangepwd.php 页面，在该页面中完成对密码的更改。首先判断用户是否非法登录，然后判断用户输入的原密码是否正确，如果正确则将用户密码更改成新值，反之给出错误提示，其代码如下：

```
<?php
require_once 'header.php';
if (! isset($_SESSION['unc']) || $_SESSION['unc'] == '') {                    //判断用户是否已经登录
    echo "<script>alert('禁止非登录！');window.location.href='" . $util->baseUrl() . "/index.html';</script>";
    exit();
}
if (isset($_POST['oldpwd']) && $_POST['oldpwd'] != '') {                      //判断用户是否已经提交了表单
    if (! $adminDB->executeSQL("select id, usernc from tb  user where usernc='" . $  SESSION['unc'] . "' and pwd='" . md5($  POST['oldpwd']) . "'",
$connID)) {
        echo "<script>alert('原密码输入有误！');</script>";                    //验证原密码是否正确
    } else {
        if ($adminDB->executeSQL("update tb  user set pwd='" . md5($  POST['pwd1']) . "', truepwd='" . $  POST['pwd1'] . "' where usernc='" .
$_SESSION['unc'] . "'", $connID)) {                                          //执行更改密码操作
            echo "<script>alert('密码更改成功！');</script>";
        } else {
            echo "<script>alert('密码更改失败！');</script>";
        }
    }
}
$smarty->display('usercenterchangepwd.phtml');
require_once 'footer.php';
```

（3）意见反馈。

用户在参与本站活动时，如果对网站的管理或运营存在建议或意见，可以通过用户中心的意见反馈模块将个人意见提交给网站管理者。明日科技图书网的意见反馈表单如图 15.23 所示。

用户填写的反馈意见将被提交到 userfeedback.php 页面，在该页面中应用数据库管理类中的 executeSQL()方法将反馈意见添加到 tb_feedback 数据表中，其代码如下：

```
if (! isset($_SESSION['unc']) || $_SESSION['unc'] == '') {                    //判断用户是否已经登录
    echo "<script>alert('禁止非登录！');window.location.href='" . $util->baseUrl() . "/index.html';</script>";
    exit();
}
if (isset($_POST['title']) && $_POST['title'] != '') {                        //判断用户是否已经提交了表单
    if ($adminDB->executeSQL("insert into tb_feedback(title, content, addtime, usernc) values('" . $_POST['title'] . "', '" . $_POST['content'] . "', '" .
date('Y-m-d H:i:s') . "', '" . $_SESSION['unc'] . "')", $connID)) {          //保存用户的反馈意见信息
        echo "<script>alert('您的反馈信息已经成功保存，谢谢您的支持！');</script>";
    } else {
        echo "<script>alert('反馈信息保存失败，请重试！');</script>";
    }
}
```

图 15.22　用户密码修改表单

图 15.23　用户意见反馈表单

秘笈心法

心法领悟 596：用户中心功能总结。

在用户中心功能模块中应用的技术可以划分为两部分：第一部分，在 Smarty 模板页中创建 form 表单，提交用户需要更新的数据；第二部分，在 PHP 动态页中，应用数据库管理类中的 executeSQL()方法执行 select、insert 和 update 语句，完成对用户的验证、数据的添加和更新操作。

实例 597	我的订单 光盘位置：光盘\MR\15\597	高级 趣味指数：★★★★☆

实例说明

明日科技图书网中，“我的订单”只是用户中心功能模块中的一个子模块，其功能是输出当前登录用户的订单信息，同时可以查看订单的详细信息，执行取消订购和确认收货的操作，其运行效果如图 15.24 所示。

图 15.24　我的订单功能展示

关键技术

在开发“我的订单”功能模块时，应用的技术如下：

（1）数据库的连接、操作类。

（2）Smarty 模板引擎的配置类。

（3）字符串的处理类，baseUrl()方法获取网址的相对路径，moneyFormat()方法对货币金额进行格式化，unHtml()方法对 html 字符串进行转义。

（4）应用 Smarty 模板中的 ldelim 和 rdelim 标签输出 Smarty 模板中的定界符，实现直接在 Smarty 模板页中定义 JavaScript。

上述技术在前面的章节中进行过详细的讲解，这里就不再赘述。

设计过程

（1）在“我的订单”功能模块中，所有的操作都在 usercenterorder.php 文件中完成，包括当前用户订单信息的循环输出、订单详细信息的查看、订单的取消以及确认收货等，其代码如下：

```
<?php
require_once 'header.php';                                    //包含网页头文件
if (!isset($_SESSION['unc']) || $_SESSION['unc'] == '') {     //判断当前用户的权限
    echo "<script>alert('禁止非登录！');window.location.href='".$util->baseUrl()."/index.html';</script>";
    exit();
}
if(isset($_GET['def']) && $_GET['def'] =='y'){                //如果变量 def 的值是 y，则执行确认收货的操作
    $adminDB->executeSQL("update tb_order set issh=1 where orderno='".$_GET['orderno']."'", $connID);
}
if(isset($_GET['def']) && $_GET['def'] =='qx'){               //如果变量 def 的值是 qx，则执行取消订单的操作
    $adminDB->executeSQL("update tb_order set isqx=1 where orderno='".$_GET['orderno']."'", $connID);
}
$isShow = 'F';                                                //初始化变量值为 F
```



```
</div>
{/if}
```

秘笈心法

心法领悟 597：实现退出登录功能。

用户在登录成功后，可以修改用户注册信息、修改用户登录密码、查看用户订单信息和提交意见反馈等，那么如何实现退出登录呢？

实现退出登录功能应用的是 session_start()、session_unregister()、unset()和 header()函数。首先，初始化 SESSION 变量，然后注销 SESSION 变量 unc，接着删除 SESSION 变量，最后跳转到 index.html 页面，该功能存储于 logout.php 文件中。

实例 598 找回密码

光盘位置：光盘\MR\15\598

高级

趣味指数：★★★★☆

实例说明

在明日科技图书网的用户登录页面中，有一个“找回密码”超链接，单击该链接将执行密码找回的操作。第一步：输入用户的昵称；第二步：输入密码提示答案；第三步：如果输入答案不正确，则给出错误提示信息，否则弹出密码找回成功对话框，并且显示原密码，也可以选择重新设置密码，其运行效果如图 15.25 所示。



图 15.25 找回密码的操作流程

关键技术

在找回密码功能实现过程中，应用到的技术如下：

（1）应用 Smarty 模板中的 ldelim 和 rdelim 标签输出 Smarty 模板中的定界符，实现直接在 Smarty 模板页中定义 JavaScript。

（2）应用 Window 对象中 open()方法打开 openfindpwd.php 文件，开始执行找回密码的操作。

（3）应用 PHP 中提供的 MySQL 数据库函数操作 MySQL 数据库，完成找回密码、更改密码的操作。其中，首先在 conn.php 文件中应用 mysql_connect()、mysql_select_db()和 mysql_query()函数完成与数据库服务器、数据库的连接，并且设置编码格式，其代码如下：

```
<?php
$conn=mysql_connect("localhost","root","111");          //连接数据库服务器
mysql_select_db("db_database15",$conn);                 //连接数据库
mysql_query("set names gb2312");                        //设置数据库编码格式
?>
```

设计过程

（1）在模板页 login.phtml 中应用 Smarty 模板中的 ldelim 和 rdelim 标签直接编写 JavaScript 脚本方法 openfindpwd()，打开 openfindpwd.php 文件，其关键代码如下：

```
<script language="JavaScript" type="text/javascript">
function openfindpwd()
{ldelim}
window.open("openfindpwd.php","newframe","left=200,top=200,width=200,height=100,menubar=no,toolbar=no,location=no,scrollbars=no,location=no
");{rdelim}
</script>
<a href="javascript:openfindpwd()" class="a1">[找会密码]</a>
```

（2）新建 openfindpwd.php 文件，创建 form 表单，提交要找回密码的用户的昵称，并且通过 JavaScript 脚

本判断用户提交的数据是否为空，将数据提交到 findpwd.php 页面。

（3）新建 findpwd.php 文件。首先，包含数据库连接文件 conn.php，然后判断用户提交的昵称在数据库中是否存在，如果存在则输出此用户注册时填写的密码提示问题，并且通过 form 表单将问题的答案提交到 showpwd.php 文件中进行处理。

（4）新建 showpwd.php 文件，首先包含数据库的连接文件 conn.php，然后判断用户提交的问题答案是否正确，如果不正确则给出提示信息，并跳转到上一页；如果正确则输出显示原密码和重设密码的超链接。

（5）新建 changepwd.php 文件，创建重新设置密码的 form 表单，将重新设置的密码提交到 savechangepwd.php 文件中。

（6）新建 savechangepwd.php 文件，完成用户密码的更新，其代码如下：

```
<?php
include_once ("conn.php");                    //连接数据库
$truepwd = $_POST[userpwd1];                  //获取提交的新密码
$pwd = md5($truepwd);                         //对密码进行 md5 加密
if (mysql_query("update tb_user set pwd='$pwd',truepwd='$truepwd' where id='" . $_POST["userid"] . "'", $conn)) {   //执行更新操作
    echo "<script>alert('密码更改成功!');history.back();</script>";
} else {
    echo "<script>alert('密码更改失败!');history.back();</script>";
}
mysql_close($conn);                           //关闭数据库连接
?>
```

秘笈心法

心法领悟 598：MySQL 数据库函数。

在明日科技图书网站项目中，为了体现本章内容的综合性，不但应用 Smarty 模板和 ADODB 类库技术，同时也融入了 MySQL 函数操作 MySQL 数据库的方法。有关通过 MySQL 函数操作 MySQL 数据库的详细讲解可以参考第 8 章的内容。

15.4　数据的输出

本节将介绍数据输出功能的实现，包括图书导航、图书分类、特别图书、图书试读、图书详细信息和新闻公告等功能。

实例 599	图书导航 光盘位置：光盘\MR\15\599	高级 趣味指数：★★★★☆

实例说明

明日科技图书网站中，通过图书导航功能展示出网站中图书的所有类别，并且将图书划分为初级、中级和高级 3 种类型，图书导航功能的运行效果如图 15.26 所示。



图 15.26　图书导航功能

关键技术

实现图书导航功能应用的关键技术如下：

（1）在 index.php 动态页中，应用数据库管理类中的 executeSQL()方法查询出图书大类和小类数据表中的数据。其中关于数据库的连接和操作方法可以参考本章实例 586 和实例 587。

（2）在 index.phtml 模板页中，应用 Smarty 模板中的 if 语句根据模板变量传递的值判断应具体输出哪种类型的数据，然后应用 section 语句进行嵌套，完成图书大类与小类的对应循环输出。

有关 Smarty 模板中 if 语句和 section 语句的详细讲解可以参考第 14 章的内容。

设计过程

（1）在 index.php 动态页中，首先判断变量$cctype 的值，如果为空，则赋值为 1，否则直接将变量的值赋给模板变量。然后，应用数据库管理类中的 executeSQL()方法查询 tb_bigtype 和 tb_smalltype 表中存储的图书类别数据，同样将查询结果赋给模板变量，其关键代码如下：

```
if(!isset($_GET['cctype']) || $_GET['cctype']==''){                    //判断变量值是否为空
        $cctype='1';                                                    //为变量赋值
}else {
        $cctype = $_GET['cctype'];
}
$smarty->assign('cctype', $cctype);                                     //将变量值赋给模板变量
//查询图书大类中的数据
$bigtypes = $adminDB->executeSQL("select id, typename from tb_bigtype order by addtime", $connID);
$smarty->assign('bigtypes', $bigtypes);
//查询图书小类中的数据
$smalltypes = $adminDB->executeSQL("select id, typename, bigtypeid from tb_smalltype order by addtime", $connID);
$smarty->assign('smalltypes', $smalltypes);
```

（2）在 index.phtml 模板页中，根据模板变量的值判断输出全部图书导航、初级入门类图书、提供必备类图书还是高级导向类图书，然后通过 section 语句嵌套循环输出所有图书类别，其关键代码如下：

```
{section name=bID loop=$bigtypes}
<div style="width:195px">
<div style="width:100%; height:25px; text-align:left; padding-left:6px; padding-top:5px; color:#990000">
        · {util->unHtml p=$bigtypes[bID].typename}
</div>
<div style="width:100%">
{section name=sID loop=$smalltypes}
        {if $bigtypes[bID].id == $smalltypes[sID].bigtypeid}
           <div style="width:49%; height:18px; float:left; text-align:left; padding-left:12px">
                · <a href="{util->baseUrl}/listbook-{$smalltypes[sID].id}-{if $cctype==1}1{elseif $cctype==2}2{elseif $cctype==3}3{elseif
$cctype==4}4{/if}.html" class="a4">{util->unHtml p=$smalltypes[sID].typename}</a>
        </div>
        {/if}
{/section}
      </div>
</div>
{/section}
```

秘笈心法

心法领悟 599：section 语句的嵌套循环。

section 语句的嵌套循环，需要 if 语句的辅助才能够完成。首先，执行 section 语句的外层循环，并输出指定的数据，然后执行 section 语句的内层循环，在内层循环中应用 if 语句进行判断，当外层循环中的指定值与内层循环中的指定值相同时，才输出内层循环中的指定数据。

在本实例中，首先循环输出模板变量$bigtypes 中的类型名称，然后执行内层循环输出模板变量$smalltypes 中的数据，此时应用 if 语句进行判断，只有当$bigtypes 中的 id 与$smalltypes 中的 bigtypeid 值相同时，才输出$smalltypes 中的类型名称，并且为这个类型名称创建超链接。



实例 600	图书分类 光盘位置：光盘\MR\15\600	高级 趣味指数：★★★★☆

实例说明

明日科技图书网是一个专业的 IT 类图书展示和销售平台，为了方便读者选购，特将图书信息按类别展现出来。用户在选购图书时，首先选择图书大类，然后在大类中选择图书的小类，这样就可以将该类别的所有图书

以分页的形式展现出来。明日科技图书网图书分类展示效果如图 15.27 所示。

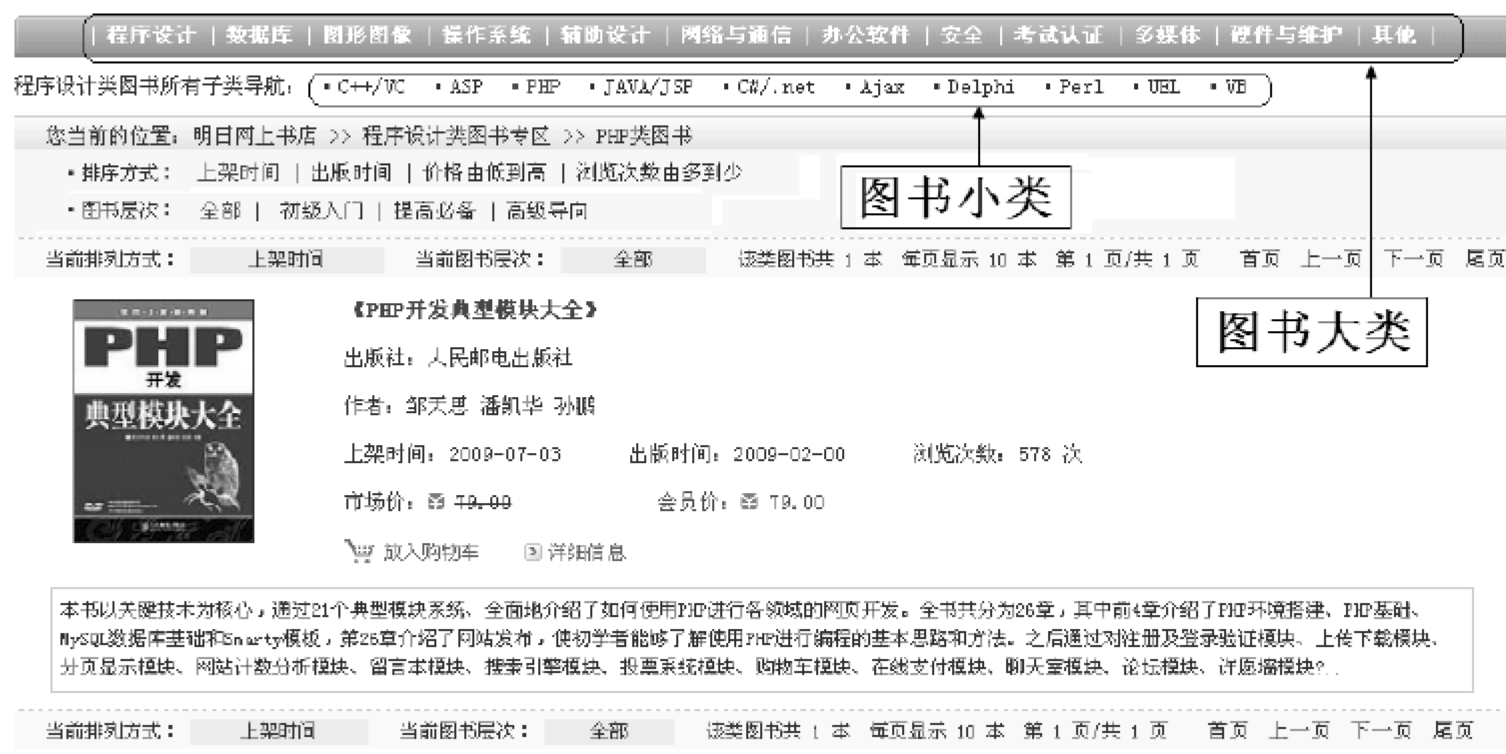


图 15.27　明日科技图书网图书分类展示效果

关键技术

实现图书分类展示的关键是数据表的创建，在图书大类表 tb_bigtype 和图书小类表 tb_smalltype 之间存在一个关联的字段，通过该字段确立两个表中数据的从属关系。这里将 tb_bigtype 表中的 id 字段与 tb_smalltype 表中的 bigtypeid 字段建立关联，即在 tb_smalltype 表中，根据字段 bigtypeid 的值可以确定该小类属于哪个大类。

设计过程

在图书分类展示页面中，浏览者可以从最上面的图书大类中选择要购买图书的大类，然后在子类中选择具体的语言类别，这样就可以将指定语言类别的图书展示出来，同时可以按照指定的排列方式和图书层次对图书的显示顺序进行调整。具体实现过程如下：

（1）在 listbook.php 文件中，首先使用$_GET 全局数组接收传递到页面的排序方式标识、图书层次标识和当前页显示的页码，然后使用分页类的 pageData()方法获取分页参数，最后指定模板页，其关键代码如下：

```
if (! isset($_GET['pltype']) || $_GET['pltype'] == '') {                    //获得图书排列方式
      $pltype = '1';
} else {
      $pltype = $_GET['pltype'];
}
$smarty->assign('pltype', $pltype);
if (! isset($_GET['cctype']) || $_GET['cctype'] == '') {                    //获得图书层次标识
      $cctype = '1';
} else {
      $cctype = $_GET['cctype'];
}
$smarty->assign('cctype', $cctype);
if (! isset($_GET['page']) || $_GET['page'] == '') {                        //获得当前要显示的页码
      $page = '1';
} else {
      $page = $_GET['page'];
}
$smarty->assign('page', $page);
$smalltype = $adminDB->executeSQL("select id, bigtypeid, typename from tb_smalltype where id='$stid'", $connID);          //查询图书小类
$smarty->assign('smalltype', $smalltype);
$bigtype = $adminDB->executeSQL("select id, typename from tb_bigtype where id='" . $smalltype[0]['bigtypeid'] . "'", $connID);
$smarty->assign('bigtype', $bigtype);
$smalltypes = $adminDB->executeSQL("select id, typename from tb_smalltype where bigtypeid='" . $bigtype[0]['id'] . "'", $connID);   //查询图书大类
$smarty->assign('smalltypes', $smalltypes);
if ($cctype == '1') {                                                       //设置图书层次条件
      $where = " and smalltypeid = '" . $stid . "'";
} elseif ($cctype == '2') {
      $where = " and bookcc = 1 and smalltypeid = '" . $stid . "'";
```

```
} elseif ($cctype == '3') {
    $where = " and bookcc = 2   and smalltypeid = '" . $stid . "'";
} elseif ($cctype == '4') {
    $where = " and bookcc = 3   and smalltypeid = '" . $stid . "'";
}
if ($pltype == '1') {                                               //设置图书排列方式
    $order = "addtime desc";
} elseif ($pltype == '2') {
    $order = "pubtime desc";
} elseif ($pltype == '3') {
    $order = "newprice asc";
} elseif ($pltype == '4') {
    $order = "browsertime desc";
}
$sql = "select tb_bookinfo.id, tb_bookinfo.bookimg, tb_bookinfo.oldprice, tb_bookinfo.newprice, tb_bookinfo.about, tb_bookinfo.browsertime,
tb_bookinfo.bookname, tb_bookinfo.writer, tb_bookinfo.pubtime, tb_bookinfo.addtime, tb_bookinfo.ishave, tb_pub.pubname from tb_bookinfo, tb_pub
where tb_bookinfo.pubid = tb_pub.id " . $where . " order by " . $order . "";         //构建查询语句
$bookinfos = $pageDB->pageData($sql, $connID, 10, $page);                           //执行分页查询
$smarty->assign('bookinfos', $bookinfos);
```

（2）在 views\listbook.phtml 模板页中，应用 section 语句完成指定类别图书信息的分页显示，关键代码如下：

```
{if $bookinfos.countRs > 0}
{section name=pbID loop=$bookinfos.data}
<div style="width:930px">
    <br>
    <div style="width:100%; height:180px">
        <div style="width:20%; height:100%; float:left">
            <a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{$system[0].bookimgurl}/
{$bookinfos.data[pbID].bookimg}" width="110" height="150" style="border:1px solid #123456" /></a>
        </div>
        <div style="width:78%; height:100%; float:right; text-align:left; font-size:13px">
            <li style="display:inline"><a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html" class="a5"><strong>
{$bookinfos.data[pbID].bookname}</strong></a></li>
                <br><br>
                <li style="display:inline">出版社：{$bookinfos.data[pbID].pubname}</li>
                <br><br>
                <li style="display:inline">作者：{$bookinfos.data[pbID].writer}</li>
                <br><br>
                <li style="display:inline">上架时间：{$bookinfos.data[pbID].addtime|truncate:10:""}  出版时间：
{$bookinfos.data[pbID].pubtime|truncate:10:""}   浏览次数：{$bookinfos.data[pbID].browsertime} 次</li>
                <br><br>
                <li style="display:inline">市场价：<img src="{util->baseUrl}/img/mark_green.gif" /> <s>{util->moneyFormat
p=$bookinfos.data[pbID].oldprice}</s>  <font color="#FF0000">会员价：<img src="{util->baseUrl}/img/mark_red.gif"
/> {util->moneyFormat p=$bookinfos.data[pbID].newprice}</font>
                 </li>
                 <br><br>
                 <li style="display:inline"><a href="{util->baseUrl}/cart-{$bookinfos.data[pbID].id}-add.html"><img src="{util->baseUrl}/
img/button_order.gif" border="0"/></a> <a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{util->baseUrl}/img/
button_info.gif" border="0"/></a></li>
        </div>
      </div>
    <div style="width:95%; height:50px; border:1px solid #CCCCCC; text-align:left; line-height:18px; color:#990000; padding:5px">
     {util->unHtml p=$bookinfos.data[pbID].about|truncate:420:"...":false}
    </div>
</div>
{/section}
{else}
<div style="width:500px; height:80px; font-size:14px; color:#FF0000">
    <img src="{util->baseUrl}/img/mark_telllogin.gif"><br />
                <strong>对不起，暂无该类图书！</strong>
</div>
{/if}
```

秘笈心法

心法领悟 600：为什么编辑的模板页文件后缀是.phtml，而输出时文件的后缀是.html。

因为本项目的开发应用 PHP 伪静态技术，会对输出文件的后缀和传递的参数进行重新定义。伪静态技术的执行要通过.htaccess 文件。

实例 601	特别图书 光盘位置：光盘\MR\15\601	高级 趣味指数：★★★★☆

实例说明

为了满足不同用户的需要，在制作明日科技图书网时，设置了特别图书展示模块，该模块可以对新书、特价图书、热卖图书和期书进行分页展示，其运行效果图如图 15.28 所示。



图 15.28　特别图书分页展示

关键技术

在特别图书模块中，包括新书推荐、特价图书、热卖图书、期书和试读 5 个子选项，在这 5 个子模块之间的跳转根据的是超链接参数 t 传递的变量值，在动态文件 listsepbook.php 中应用 if 语句进行判断，并且根据判断的结果为模板变量赋值，然后在模板变量中输出对应的内容。listsepbook.php 文件的关键代码如下：

```
if ($_GET['t'] == 'new') {
    $nowtype = '新书推荐';
    $andwhere = "isnew = 1";
} elseif ($_GET['t'] == 'sepprice') {
    $nowtype = '特价图书';
    $andwhere = "issepprice = 1";
} elseif ($_GET['t'] == 'hotsell') {
    $nowtype = '热卖图书';
    $andwhere = "ishotsell = 1";
} elseif ($_GET['t'] == 'term') {
    $nowtype = '期书';
    $andwhere = "isterm = 1";
} elseif ($_GET['t'] == 'mrbook') {
    $nowtype = '明日图书推荐';
    $andwhere = "ismrbooktj = 1";
}
$smarty->assign('gt', $_GET['t']);
$smarty->assign('nowtype', $nowtype);
```

在完成特别图书的分页输出时，编写一个多表查询的 SQL 语句，对 tb_bookinfo 和 tb_pub 两个数据表进行查询，其使用的 SQL 语句如下：

```
$sql = "select tb_bookinfo.id, tb_bookinfo.bookimg, tb_bookinfo.about, tb_bookinfo.browsertime, tb_bookinfo.oldprice, tb_bookinfo.newprice, tb_bookinfo.bookname, tb_bookinfo.writer, tb_bookinfo.pubtime, tb_bookinfo.addtime, tb_bookinfo.ishave, tb_pub.pubname from tb_bookinfo, tb_pub where tb_bookinfo.pubid = tb_pub.id and " . $andwhere . " order by tb_bookinfo.addtime desc";
```

设计过程

（1）建立 listsepbook.php 文件用来实现业务逻辑。在该文件中，首先使用$_GET 全局数组接收查询字符串参数 t 的值，然后根据该值决定当前页面要显示的内容，并构建查询语句，最后使用分页类的 pageData()方法实现分页查询，其完整代码请参考本书光盘，这里不再赘述。

（2）建立模板页 listsepbook.phtml，通过 section 语句分页输出指定类别下的图书信息，其关键代码如下：

```
{if $bookinfos.countRs > 0}
{section name=pbID loop=$bookinfos.data}
<div style="width:930px">
      <br>
      <div style="width:100%; height:180px">
            <div style="width:20%; height:100%; float:left">
                  <a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{$system[0].bookimgurl}/
{$bookinfos.data[pbID].bookimg}" width="110" height="150" style="border:1px solid #123456" /></a>
            </div>
            <div style="width:78%; height:100%; float:right; text-align:left; font-size:13px">
                  <li style="display:inline"><a href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html" class="a5"><strong>
{$bookinfos.data[pbID].bookname}</strong></a></li>
                  <br><br>
                  <li style="display:inline">出版社：{$bookinfos.data[pbID].pubname}</li>
                  <br><br>
                  <li style="display:inline">作者：{$bookinfos.data[pbID].writer}</li>
                  <br><br>
                  <li style="display:inline">上架时间：{$bookinfos.data[pbID].addtime|truncate:10:""}      出版时
间：{$bookinfos.data[pbID].pubtime|truncate:10:""}      浏览次数：{$bookinfos.data[pbID].browsertime} 
次</li>
                  <br><br>
                  <li style="display:inline">市场价：<img src="{util->baseUrl}/img/mark_green.gif" /> <s>{util->moneyFormat
p=$bookinfos.data[pbID].oldprice}</s>            
                        <font color="#FF0000">会员价：<img src="{util->baseUrl}/img/mark_red.gif" /> {util->moneyFormat
p=$bookinfos.data[pbID].newprice}</font>
                   </li>
                   <br><br>
                   <li style="display:inline"><a href="{util->baseUrl}/cart-{$bookinfos.data[pbID].id}-add.html"><img
src="{util->baseUrl}/img/button_order.gif" border="0"/></a>    <a
href="{util->baseUrl}/bookinfo-{$bookinfos.data[pbID].id}.html"><img src="{util->baseUrl}/img/button_info.gif" border="0"/></a></li>
            </div>
                  </div>
      <div style="width:95%; height:50px; border:1px solid #CCCCCC; text-align:left; line-height:18px; color:#990000; padding:5px">
       {util->unHtml p=$bookinfos.data[pbID].about|truncate:420:"...":false}
      </div>
</div>
{/section}
{else}
<div style="width:500px; height:80px; font-size:14px; color:#FF0000">
      <img src="{util->baseUrl}/img/mark_telllogin.gif"><br />
                  <strong>对不起，暂无新书信息！</strong>
</div>
{/if}
```

秘笈心法

心法领悟 601：SQL 语句中的多表查询。

在 SQL 语言中，通过 where 子句实现多表查询，所要查找的字段名用“表明.字段名”表示，这样可以避免因表之间字段重名而无法获知该字段属于哪个表。在 where 子句中多个表之间所形成的联动关系应按如下格式书写：

```
表 1.字段=表 2.字段  and  其他查询条件
```

多表查询的语法如下：

```
select  字段名  from  表 1,表 2……  where  表 1.字段=表 2.字段  and  其他查询条件
```

实例 602	图书试读 光盘位置：光盘\MR\15\602	高级 趣味指数：★★★★☆

实例说明

为了能够让读者了解某本图书的大概内容，吸引更多读者购买图书，在制作明日科技图书网时，将图书试读作为单独模块，其运行效果如图 15.29 所示。



图 15.29　试读图书展示

关键技术

实现图书试读的关键技术是如何获取到试读文件，本实例中应用 HTTP 方式完成试读文件的下载操作，其中关键是 header()函数的应用。有关应用 HTTP 方式进行下载的详细讲解请参考本书第 4 章的内容。

在本项目中，下载的操作在 upfiles\doc\download.php 文件中完成，同时还对文件被下载的次数进行统计，其关键代码如下：

```
<?php
require_once 'lzh.inc.php';                    //包含类的实例化文件
$filename=$_GET['filename'];                   //获取文件名称
$rid = $_GET['rid'];                           //获取文件的 id
$system = $adminDB->executeSQL("update tb_read set downtimes = downtimes+1 where id='$rid'", $connID);        //执行更新操作
if(file_exists($filename)==false){             //判断上传文件是否存在
 echo "<script>alert('对不起，本站暂时停止该文件下载!');history.back();</script>";
 exit;
}
$fp=fopen($filename,"r");                      //打开上传文件
header("Content-type:application/octet-stream");
header("Accept-ranges:bytes");
header("Accept-length:".filesize($filename));
header("Content-Disposition:attachment;filename=".$filename);
echo fread($fp,filesize($filename));           //读取上传文件内容
fclose($fp);                                   //关闭文件
```

设计过程

图书试读模块对明日科技图书网站中所有可以试读的图书进行分页展示。

（1）新建 listreadbook.php 文件，完成所有试读图书的分页查询操作。首先，通过$_GET 全局数组接收查询字符串 page 的值，该值用来指定当前页面要显示的内容。然后，构建查询语句，通过分页类的 pageData()方法获得分页数据。最后，将查询结果赋给模板变量，并且指定模板页，具体实现代码如下：



```
    //设置页面的最后更新日期(用格林威治时间表示)，使浏览器获取最新资料
    header("Last-Modified: " . gmdate("D, d M Y H:i:s") . "GMT");
    header("Cache-Control: no-cache,     must-revalidate");          //控制页面不使用缓存
    header("Pragma: no-cache");                                      //参数与以前的服务器兼容，即兼容 HTTP1.0 协议
    header("Content-type:     application/file");                    //输出类型
    header("Content-Length:      227685");                           //文件长度
    header("Accept-Ranges:      bytes");                             //接受的范围单位
    //缺省时文件保存对话框中的文件名称
    header("Content—Disposition: attachment; filename=$filename");   //实现下载
?>
```

（3）输出状态值到浏览器，控制访问权限。

```
<?php
 header('HTTP/1.1 401 Unauthorized');
 header('status: 401 Unauthorized');
?>
```

（4）实现 HTTP 下载。

```
header("Content-type: application/x-gzip");
header("Content-Disposition: attachment; filename=文件名");
header("Content-Description: PHP3 Generated Data"); >
```

HTTP 标头有很多，这里只介绍下载的 HTTP 标头，其代码如下：

```
header('Content-Disposition: attachment; filename="filename"');
```

在应用过程中，唯一需要改动的就是 filename，即将 filename 替换为要下载的文件。

实例 603　图书详细信息展示

光盘位置：光盘\MR\15\603

高级

趣味指数：★★★★☆

实例说明

为了让浏览者能够更加细致地了解所查询图书的详细信息，并能够对所选购的图书进行评论，还需要通过图书详细信息页将图书信息展示出来，效果如图 15.30 所示。



图 15.30　图书详细信息展示

关键技术

图书详细信息展示页中应用的技术如下：

（1）数据库的连接、操作类。



```
tb_bookinfo.smalltypeid = tb_smalltype.id and tb_bookinfo.id = '" . $bid . "'", $connID);    //获得图书信息
$smarty->assign('bookinfo', $bookinfo);
```

代码说明：

❶ 这里将用户所浏览过的图书 ID 用字符“@”连接保存到 SESSION 中。

❷ 函数 array_push()用来将元素添加到指定的数组中。

（2）为了帮助读者选购套系图书，制订学习流程，在制作明日科技图书网的图书详细信息页时，增加图书组合购买功能，其效果如图 15.31 所示。



图 15.31　图书组合购买

在数据库中使用 tb_book 表的 bookids 字段将组合购买图书的 ID 用字符“@”连接，在 bookinfo.php 页面中，首先使用函数 explode()将组合 ID 保存到数组中，然后使用 for 循环语句遍历该数组，根据该数组元素查询出所有组合图书信息并将图书信息保存到$arrayZhBooks 数组中，最后将该数组传递给模板，在模板中通过 section 语句输出所有组合图书信息，其关键代码如下：

```
$arrayBookids = explode('@', $bookinfo[0]['bookids']);          //使用字符“@”分割组合图书 ID
$arrayZhBooks = array();
$zhIDs = $bookinfo[0]['bid'] . '@';
for ($i = 0; $i < count($arrayBookids); $i ++) {                //获得组合 ID 图书信息
      if ($arrayBookids[$i] != '') {
            $tmpBookinfo = $adminDB->executeSQL("select id, bookname, bookimg from tb_bookinfo where id='" . $arrayBookids[$i] . "'", $connID);
            array_push($arrayZhBooks, $tmpBookinfo);
            $zhIDs .= $tmpBookinfo[0]['id'] . '@';
      }
}
$smarty->assign('zhIDs', $zhIDs);
$smarty->assign('arrayZhBooks', $arrayZhBooks);
```

（3）在图书详细信息页加入读者评论功能，有助于网站经营者分析图书的价值，从而确定图书在站点中的摆放位置，实现评论功能的过程如下：

① 在图书详细信息展示的模板页中创建 form 表单，将用户评论提交到 bookinfo.php 页面，添加到指定的数据表中。

② 在 bookinfo.php 页面中读取图书评论信息，并且将评论信息在模板页 bookinfo.phtml 中输出。同时在模板页中为图书评论信息创建“删除”超链接，使评论的发布者可以删除自己的评论。

本实例的完整代码请参考本书光盘，由于篇幅所限，这里不再赘述。

秘笈心法

心法领悟 603：内容简介和图书目录之间的切换。

在图书详细信息展示的模板页中，通过 onmouseover 事件调用 JavaScript 脚本中的 changeTab1()方法实现在内容简介和图书目录之间的切换，其原理是通过 onmouseover 事件调用 JavaScript 脚本中的方法，控制<div>标签样式的隐藏和展开。

应用的过程中，唯一需要改动的就是 filename，即将 filename 替换为要下载的文件。

实例 604	新闻公告 光盘位置：光盘\MR\15\604	高级 趣味指数：★★★★☆

实例说明

在明日科技图书网的主页中有一个新闻公告展示模块，用于展示明日科技网站发布的新闻公告信息，其运行效果如图 15.32 所示。

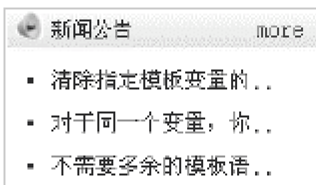


图 15.32　新闻公告



在本实例中，运用了两种截取中文字符串的方法。第一种是 Smarty 模板中的 truncate 方法，它虽然可以对中文字符串进行截取，但是同样可能会出现乱码的问题；第二种是自定义的方法 msubstr()，可以防止截取中文字符串时出现乱码，在使用时需要传递 3 个参数，指定截取的字符串、截取的开始位置和截取的长度。

15.5　站 内 搜 索

搜索功能是一个很重要且常见的功能。一个好的搜索引擎能给用户的使用带来方便。本节将以明日科技图书网站中的一般搜索和高级搜索为例，讲解搜索功能的实现方法。

实例 605	一般搜索 光盘位置：光盘\MR\15\605	高级 趣味指数：★★★★☆

实例说明

在明日科技图书网站的头文件中，设计了一个查找图书的文本框，其实现的就是一般搜索的功能。在本实例中，按图书名称进行搜索，输入 PHP 关键字，单击“查找图书”按钮，将输出如图 15.33 所示的内容。



图 15.33　查找图书信息

关键技术

在一般搜索功能中，关键技术是应用 explode()函数对查询的关键字进行拆分，返回一个数组。然后通过 for 循环语句读取这个数组中的元素。最后，定义 select 查询中的 where 子句，以数组中的元素值为 like 模糊查询的条件，其关键代码如下：

```
$arrayKeyWord = explode(' ', $keyWord);                    //使用空格对查询的关键字进行拆分
$where = "";                                               //定义变量
$j = 0;
for($i=0; $i<count($arrayKeyWord); $i++){                  //循环读取查询关键字数组中的元素
    if($arrayKeyWord[$i]!=''){                             //判断数组元素的值是否为空
        if($j>0){                                          //如果变量值大于 0
            $where .= " or ";                              //定义 where 变量的值
        }
        $where .= " tb_bookinfo.bookname like '%".$arrayKeyWord[$i]."%' ";   //定义变量的值，即定义查询语句的 where 子句
        $j++;
    }
}
```



如果参数 limit 是负数，则返回除最后的–limit 个元素外的所有元素。

实例 606　高级搜索　高级

光盘位置：光盘\MR\15\606　趣味指数：★★★★☆

实例说明

在明日科技图书网站的高级搜索功能中，可以输入图书名称关键字、作者、出版社、类别和出版时间，根据这些内容进行综合查询，高级搜索的运行效果如图 15.34 所示。

图 15.34　高级搜索表单

关键技术

（1）应用 Smarty 模板中的{ldelim}{rdelim}标签在 Smarty 模板页中直接编辑 JavaScript 脚本语句。

（2）在 JavaScript 脚本中应用 jQuery 技术编写图书类别展示的二级联动菜单，其关键代码如下：

```
<script src="{util->baseUrl}/js/jquery.js"></script>              <!--载入 jQuery 库-->
<script language="javascript" >
$(document).ready(
      function(){ldelim}
        $("#bigtypeid").change(function(){ldelim}
                  $.get("{util->baseUrl}/getstype.php?stid="+$("#bigtypeid").val(), null, function(data){ldelim}
                        $("#smalltypeid").empty();
                        $("#smalltypeid").append(data);
                {rdelim});
            {rdelim});
      {rdelim}
);
</script>
```

其中调用 getstype.php 文件，定义图书类别中子类的下拉列表框值，getstype.php 文件的代码如下：

```
<?php
require_once 'lzh.inc.php';                          //包含配置文件
$smalltypes = $adminDB->executeSQL("select id, typename from tb_smalltype where bigtypeid='".$_GET['stid']."'", $connID);
$data = "";
for($i = 0; $i < count($smalltypes); $i++){          //将从数据库中读取的数据添加到下拉列表的选项中
      $data .= "<option value=\"".$smalltypes[$i]['id']."\">".iconv('gbk', 'utf-8', $smalltypes[$i]['typename'])."</option>";
}
echo $data;                                          //输出变量值，即输出下拉列表的值
```

（3）应用 Smarty 模板中的 html_options()方法自定义函数设置日期的下拉列表框值，其部分代码如下：

```
<select name="fyear">
<option value='' selected>-请选择-</option>
      {html_options options = $arrayYear}
</select> 年
<select name="fmonth">
      <option value='' selected>-请选择-</option>
      {html_options options = $arrayMonth}
</select> 月 - - 到
```

（4）定义高级搜索的 SQL 语句，充分发挥 where 子句的作用，通过多表查询完成高级搜索的功能。



（4）在 search.phtml 模板页中，应用 section 语句分页输出查询结果，代码请参考本书光盘。

秘笈心法

心法领悟 606：jQuery 框架。

jQuery 由美国人 John Resig 创建，至今已吸引了来自世界各地的众多 JavaScript 高手加入其团队。jQuery 是继 prototype 之后又一个优秀的 JavaScript 框架，其宗旨是 WRITE LESS，DO MORE，即写更少的代码，做更多的事情。它是轻量级的 js 库（压缩后只有 21KB），这是其他 js 库所不及的。它兼容 CSS3 以及各种浏览器（IE 6.0+、FF 1.5+、Safari 2.0+、Opera 9.0+等）。

jQuery 是一个快速的、简洁的 javaScript 库，使用户能更方便地处理 HTML documents、events、实现动画效果，并且方便地为网站提供 AJAX 交互。

应用 jQuery 技术的前提条件是必须在项目中载入 jQuery 库，否则无法调用其中的方法。

15.6　购　物　车

购物车是网上购物时使用的一个临时存储商品的“车辆”，为用户在网上购物提供了很大的方便，不用担心一次购买多个商品时要进行多次提交结算的操作，可以将所选商品放入购物车中，等选购完成后一起结算，购物车是电子商务类网站中一个必不可少的功能。本节将以明日网上书店中的购物车为模板，详细讲解购物车的实现过程。

实例 607	购物车类 光盘位置：光盘\MR\15\607	高级 趣味指数：★★★★☆

实例说明

对于电子商务类网站来说，安全、稳定的购物车模块是非常重要的，该模块直接关系到交易是否能够顺利进行。为了提高代码的可移植性，在制作明日科技书店购物车时，将购物车抽象成类，封装到 library\Cart.php 文件中。购物车功能的运行效果如图 15.35 所示。



图 15.35　购物车的运行效果

关键技术

在购物车类的封装中，应用最多的就是 explode()函数和 implode()函数。

（1）explode()函数，将字符串用指定的子串分隔，并将分隔结果保存到数组中。在购物车类的 addCart()方法中，应用 explode()函数将 ID 串用@分隔，并将分隔结果保存到数组中，其关键代码如下：

```
$arrayIds = explode('@', $this->idStr);                    //将 ID 串用@分隔，并将分隔结果保存到数组中
```

（2）implode()函数，将指定数组中的所有元素使用特定的字符进行连接形成新的字符串，其语法如下：



的 ID 相同的 ID 号，则使用函数 unset()将数组元素销毁，然后再将数组中所有元素重新用字符@连接成商品 ID 串，这样就可以实现将指定的商品从购物车中移除的操作。

（5）使用 changeNum()方法更改购物车中商品的数量，其代码如下：

```
function changeNum($goodsId, $goodsNum){
    $arrayIds = explode('@', $this->idStr);                    //将商品 ID 串用@分割成数组
    $arrayNums = explode('@', $this->numStr);                  //将商品数量串用@分割成数组
    for($i = 0; $i < count($arrayIds); $i++){                  //遍历商品 ID 串
        if ($arrayIds[$i] == $goodsId){                        //如果查找到要更改的 ID
                $arrayNums[$i] = $goodsNum;                    //更改数量
        }
    }
  $this->idStr = implode('@', $arrayIds);                      //重新合并成 ID 串
    $this->numStr = implode('@', $arrayNums);                  //重新合并成数量串
}
```

changeNum()方法中包含两个参数，分别为要更改的商品 ID 号和要更改的数量。在该方法内部，首先使用函数 explode()将商品 ID 串和商品数量串用@进行分割，并将分割的字串分别保存到数组中，然后通过 for 循环语句遍历商品 ID 数组，在其中查找要更改数量的商品 ID，查找到后，将新数量值赋给数量串中对应的数组元素，最后使用函数 implode()将商品 ID 数组和商品数量数组中的所有元素分别重新用字符@进行连接，组成新的商品 ID 串和商品数量串。

（6）通过 Cart 类的 setCartNull()方法，清空购物车中的商品。实现清空购物车中商品的操作，只需将商品 ID 串和商品数量串赋为空值即可，其代码如下：

```
function setCartNull(){
    $this->idStr = '';                                         //清空商品 ID 串
    $this->numStr = '';                                        //清空商品数量串
}
```

（7）通过购物车类中的方法完成对商品 ID 串和商品数量串的操作后，还需要定义两个方法，分别用来获取商品 ID 串和商品数量串，其代码如下：

```
function getIdStr(){                                           //获得购物车的商品 ID 串
    return $this->idStr;
}
function getNumStr(){                                          //获得购物车的商品数量串
    return $this->numStr;
}
```

以上讲解了明日科技书店购物车功能的 Cart 类的属性和主要方法，其存储于 library\Cart.php 文件中。

秘笈心法

心法领悟 607：解析购物车类中的商品 ID 串和数量串。

购物车类中的商品 ID 串和商品数量串是指用字符@连接的多个商品的 ID 号和数量值，其目的是使用一个变量或者数据表中的一个字段存储多个商品的 ID 或数量。

实例 608	购物车功能实现 光盘位置：光盘\MR\15\608	高级 趣味指数：★★★☆

实例说明

购物车的主要功能是保留用户选择的商品信息。用户可以在购物车内设置选购商品的数量、显示选购商品的总金额，还可以进行清除选择的全部商品信息、重新选择商品信息等操作。在实例 607 中已经讲解了购物车类的实现过程，本实例将在此基础上介绍购物车的页面设计和功能实现，其中明日科技书店的购物车运行效果如图 15.36 所示。





```
    $_SESSION['toUrl'] = 'getbuyuserinfo.html';
}
```

（8）在 views 文件夹下新建 cart.phtml 模板页，完成购物车页面的设计。

秘笈心法

心法领悟 608：判断指定的数据是否是正整数。

判断指定的数据是否是正整数，应用到 3 个函数：is_numeric()、intval()和 floor()。

is_numeric()函数，检测变量是否为数字或者数字字符串，如果是则返回 TRUE，否则返回 FALSE。

intval()函数，获取变量的整数值。

floor()函数，利用舍去法获取变量的整数值。

在 if 条件语句中，判断 is_numeric()函数的返回值是否为 TRUE，判断 intval()函数的返回值是否大于 0，判断 floor()函数的返回值与变量值是否相等。当满足 is_numeric()函数的返回值为 TRUE，intval()函数的返回值大于 0，floor()函数的返回值与变量值相等时，说明变量是正整数，否则说明不是正整数，其应用方法如下：

```
if (! is_numeric($changenum) || intval($changenum) <= 0 || floor($changenum) != $changenum) {
        echo "<script>alert('购买图书的数量只能为正整数！ ');</script>";
}
```

其中变量$changenum 为要验证的变量。

15.7　图书订单处理

为了方便用户购买图书，促进图书销售，在开发明日科技图书网时特别设立图书订单处理模块，通过该模块用户可以方便地订购图书并进行在线支付。本节将对明日科技书店图书订单处理模块进行讲解。



实例 609	填写收货人信息 光盘位置：光盘\MR\15\609	高级 趣味指数：★★★★

实例说明

为了能够将用户所购买的图书顺利邮寄到用户手中，在用户选择好购买的图书后还需要填写收货人信息，明日科技图书网用于填写收货人信息的页面如图 15.37 所示，在该页面中，用户需要填写个人的真实联系信息、收货方式等内容，如果需要发票，还需要填写发票抬头。

图 15.37　填写收货人信息的页面



```
            $recpay = '20';
        } elseif ($rect == '4') {
            $recpay = '30';
        }
        $_SESSION['recuserinfo']['recpay'] = $recpay;
        echo "<script>window.location.href='" . $util->baseUrl() . "/deforder.html';</script>";          //重新定向到 deforder.html 页面
        exit();
    }
    $smarty->display('getbuyuserinfo.phtml');
    require_once 'footer.php';
```

秘笈心法

心法领悟 609：在 Smarty 模板中直接获取 SESSION 变量的值。

在 Smarty 模板中直接获取 SESSION 变量的值，应用的是 Smarty 模板中的$smarty.session 标签。首先，在 PHP 动态页中，定义 SESSION 变量，其格式如下：

```
$_SESSION['recuserinfo']['username'] = $_POST['username'];
$_SESSION['recuserinfo']['sex'] = $_POST['sex'];
```

然后，在 Smarty 模板页中，通过如下方法获取 SESSION 变量的值：

```
$smarty.session.recuserinfo.username
$smarty.session.recuserinfo.sex
```

实例 610	确认订购信息 光盘位置：光盘\MR\15\610	高级 趣味指数：★★★★☆

实例说明

用户填写完收货人信息后将进入订购信息确认页面，如图 15.38 所示，在该页面中可以查看并确认用户所订购图书的名称、单价、总价格和收货人的信息。

图 15.38　确认订购信息

关键技术

确认订购信息功能的实现，首先应用的是 explode()函数对商品 ID 串和商品数量串中的数据进行读取，然后根据返回的数组元素值，从数据库中读取图书信息，完成图书价格的汇总。

在 deforder.phtml 模板页中，通过 section 语句循环输出用户购买的图书信息，并且通过$smarty.session 标签输出收货人的信息。上述方法都已经在前面的实例中讲解过，这里不再赘述。

设计过程

（1）为了使图书订购流程的业务逻辑更加方便，将购物车和收货人的信息保存到 session 中，所以显示用





网站中提交的数据，并且将数据赋给模板变量，在模板页中通过 form 表单将数据提交到支付宝的网站中。defpay.php 动态页的关键代码如下：

```
<?php
require_once 'header.php';
$t = $_GET['t'];
if ($t == 'qyzh') {
      $paytype = '企业账户汇款';
      $alertType = '0';
}elseif ($t == 'zfb') {
      $paytype = '支付宝在线支付';
      $alertType = '2';
} elseif ($t == 'ghzxzf') {
      $paytype = '工行在线支付';
      $alertType = '3';
}
//省略了部分代码
if ($alertType == '2') {
      $smarty->assign('orderid', $_SESSION['recuserinfo']['orderNo']);
      $smarty->assign('ydType', 'EXPRESS');
      $smarty->assign('ydPrice', $_SESSION['recuserinfo']['recpay']);
      $smarty->assign('bookPrice', $_SESSION['recuserinfo']['bookprice']);
}
```

在 defpay.phtml 模板页中，输出订单的确认信息，创建向支付宝网站中提交数据的 form 表单，判断当模板变量$alertType 的值等于 2 时，完成向支付宝网站中提交数据的操作，其关键代码如下：

```
<form name="form_zfb" method="post" action="http://221.8.65.77/zfb/index.php" target="_blank">
      <input type="hidden" name="ddno" value="{$orderid}">
      <input type="hidden" name="ydType" value="{$ydType}">
      <input type="hidden" name="ydPrice" value="{util->moneyFormat p=$ydPrice}">
      <input type="hidden" name="price" value="{util->moneyFormat p=$bookPrice}">
</form>
<a {if $alertType == '0' }
href="javascript:alert('您的订购信息已经保存,请到指定银行汇款!');"
{elseif $alertType == '1'}
      href="javascript:alert('您的订购信息已经保存,请到邮局邮款!');"
{elseif $alertType == '2'}
        href="javascript:form_zfb.submit();"
{elseif $alertType == '3'}
        href="javascript:toicbc.submit();"
{/if}
><img src="{util->baseUrl}/img/button_defpay.gif" border="0" /></a>
```

秘笈心法

心法领悟 611：支付宝在线支付。

声明：这里讲解的是通过 PHP 实现电子商务网站与支付宝进行交互的方法，其中涉及到的支付宝提供的参数和设置都是虚拟的，真实的参数将在实际的运作中由支付宝提供。

实例 612	工行在线支付 光盘位置：光盘\MR\15\612	高级 趣味指数：★★★★☆

实例说明

在选择支付方式页面中，单击工行在线支付的“支付”超链接，将跳转到如图 15.42 所示的页面，输出本次图书订购的订单号、支付方式和需要支付的总金额。

如果单击图 15.42 中的“已经确认订购信息按选择支付方式支付”按钮，将跳转到工行在线支付页面，如图 15.43 所示。

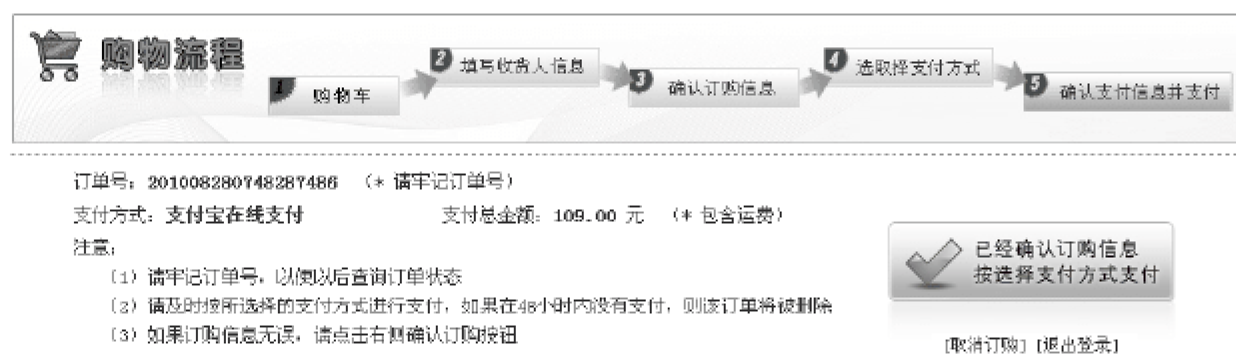


图 15.42　支付宝在线支付



图 15.43　工行在线支付

关键技术

工行在线支付功能的实现主要根据工行提供的在线支付的接口，将订单中的数据以表单的形式提交到工行指定的网址中。详细的参数设置可以参考工行提供的文档说明书和实例程序。

在实现与工行接口的连接之前，首先要从工行提供的说明书中找到下面两个文件：开发 API 接口\LIB\windows\win32\infosecapi.dll（XP 系统使用的文件）和开发 API 接口\ASP\ ICBCEBankUtil.dll，并将其复制到系统文件夹下的 WINDOWS\system32 下，在运行命令中输入 regsvr32 ICBCEBankUtil.dll，加载服务。

通过 PHP 实现网站与工行接口交互的方法如下：

首先，通过 com 类中的 init()方法对订单签名数据和商城的公钥进行加密处理，并且将指定的数据以表单的形式提交到工行指定的页面中，其关键代码如下：

```
$src="ICBC  PERBANK  B2C1.0.0.0".$infomer["merid"].$infomer["meracct"].$returnaddress."HS".$ddnumber.$amount."0010".$nowtime."0";
$com=new com('ICBCEBANKUTIL.B2CUtil');
$rc=$com->init("c:\WINDOWS\user.crt","d: \user.crt","d: \user.key","00");
$ssrc=$com->signC($src, strlen($src));                              //订单签名数据
$rc=$com->verifySignC($src,strlen($src), $ssrc, strlen($ssrc));
$cert=$com->getCert(1);                                             //商城证书公钥
```

完成订单签名数据和商城证书公钥的加密处理后，将商户的测试证书存储到 c:\WINDOWS 目录下，设置商户的私钥口令为 00。

然后，创建 form 表单，将指定的数据提交到工行指定的网站中。

设计过程

工行在线支付的操作同样由 defpay.php 和 defpay.phtml 两个文件组成。

（1）在 defpay.php 动态页中，判断如果选择的是工行在线支付（$alertType = '3'），则定义向工行在线支付网站中提交的数据，并且将数据赋给模板变量，在模板页中通过 form 表单将数据提交到工行在线支付的网站中。在 defpay.php 动态页中，有关工行在线支付的关键代码如下：

```
<?php
require_once 'header.php';
$t = $_GET['t'];
if ($t == 'qyzh') {
      $paytype = '企业账户汇款';
      $alertType = '0';
}elseif ($t == 'zfb') {
      $paytype = '支付宝在线支付';
      $alertType = '2';
} elseif ($t == 'ghzxzf') {
      $paytype = '工行在线支付';
      $alertType = '3';
}
//省略了部分代码
if ($alertType == '3') {
      $merInfo = $adminDB->executeSQL("select merid, meracct from tb_system where mark = 1", $connID);
      $merid = $merInfo[0]['merid'];
      $meracct = $merInfo[0]['meracct'];
      $merURL = 'http://www.mrbooks.cn';
```





图 15.44　后台管理系统登录

关键技术

在后台管理系统登录功能实现过程中，应用 JavaScript 脚本调用 xym.php 文件完成验证码的输出操作。在 xym.php 文件中，应用文件系统函数库中的函数将指定的图片以二进制的形式输出到浏览器，其代码如下：

```
<?php
$num=$_GET[num];                              //获得传递的数字
$address="img/code/".$num.".gif";             //获取对应图片地址
$fp=fopen($address,"r");                      //打开对应的图片
echo fread($fp,filesize($address));           //将图片以二进制的形式输出到浏览器
fclose($fp);   //关闭打开的图片
?>
```

有关文件系统函数的应用可以参考本书第 4 章的内容，这里不再赘述。

设计过程

明日科技图书网后台登录页面的制作过程主要包括登录表单的设计和登录信息验证。

（1）后台用户登录表单的 UI 是通过 HTML 语言的 input 标签实现的，并通过 JavaScript 实现图片验证码，admin-login.phtml 的关键代码如下：

```
<form name="form_login" method="post" action="{util->baseUrl}/admin-login.php" onsubmit="return chkinputadmin(this)">
<div style="width:300px; height:200px; text-align:left">
用户名：<input type="text" name="anc" size="22" class="input" /><br /><br />
密码：<input type="password" name="pwd" size="22" class="input" /><br /><br />
验证码：<input type="text" name="xym" size="8" class="input" /><input type="hidden" value="" name="xym1">
<script language="javascript">
    var num1=Math.round(Math.random()*10000000);
    var num=num1.toString().substr(0,4);
    document.write("<img name=codeimg4 src='xym.php?num="+num.substr(0,1)+"'>");
    document.write("<img name=codeimg5 src='xym.php?num="+num.substr(1,1)+"'>");
    document.write("<img name=codeimg6 src='xym.php?num="+num.substr(2,1)+"'>");
    document.write("<img name=codeimg7 src='xym.php?num="+num.substr(3,1)+"'>");
    form_login.xym1.value=num;
    function code_1(){ldelim}
        var num1=Math.round(Math.random()*10000000);
        var num=num1.toString().substr(0,4);
        document.codeimg4.src="xym.php?num="+num.substr(0,1);
        document.codeimg5.src="xym.php?num="+num.substr(1,1);
        document.codeimg6.src="xym.php?num="+num.substr(2,1);
        document.codeimg7.src="xym.php?num="+num.substr(3,1);
        form_login.xym1.value=num;
        {rdelim}
</script>
<a href="javascript:code_1()" class="a1">看不清</a>
<input type="submit" value="登录" />  <input type="reset" value="重置" />
</div>
</form>
```

（2）在 admin-login.php 文件中完成对用户名和密码的验证，如果用户名和密码不正确，则提示登录失败；否则将登录用户名存储到 SESSION 变量中，并且跳转到后台管理系统的 admin-index.php 页面，其代码如下：

```
require_once 'lzh.inc.php';
if (isset($_POST['anc']) && $_POST['anc'] != '') {          //判断用户是否提交了表单
    if ($adminDB->executeSQL("select id, usernc from tb_user where usernc='" . $_POST['anc'] . "' and pwd='" . md5(trim($_POST['pwd'])) . "' and
usertype=1", $connID)) {                                    //根据用户昵称和密码查询管理员
        if (isset($_SESSION['anc'])) {                      //如果查询到则说明登录成功，这时将管理员昵称保存到 SESSION 中
            unset($_SESSION['anc']);
        }
        $_SESSION['anc'] = $_POST['anc'];
        echo "<script>window.location.href='" . $util->baseUrl() . "/admin-index.php';</script>";    //重定向到后台首页
    } else {
        echo "<script>alert('登录失败！');</script>";        //给出登录失败提示
    }
}
$smarty->display('admin-login.phtml');
$connDB->closeConnID();                                     //关闭数据库连接
```

秘笈心法

心法领悟 613：防止后台管理系统中的程序被非法访问。

防止后台管理系统中的程序被非法访问，关键就在于登录成功后创建的$_SESSION['anc']变量，通过这个变量值来确保后台管理员的特殊权限。在执行后台管理系统中的操作时，都会首先判断这个$_SESSION['anc']变量的值，如果为空，则说明是非法访问，那么将给出提示信息，并跳转到后台管理系统的登录页面。

实例 614	后台管理系统主页设计 光盘位置：光盘\MR\15\614	高级 趣味指数：★★★★☆

实例说明

明日科技图书网站的后台管理员登录成功后，将进入到后台管理系统的主页，主页通过 frame 框架设计，展示出后台管理系统中的所有功能，其页面效果如图 15.45 所示。



图 15.45　后台管理系统主页

关键技术

在后台管理系统的主页中，应用 frame 框架将后台管理系统主页分为两部分：左侧的导航页面和右侧的中心内容显示页面。frame 框架的代码如下：



秘笈心法

心法领悟 614：超链接中 target 属性的应用。

超链接中 target 属性包含 5 个属性值：_blank 在新窗口中打开超链接文件；_parent 将链接文件载入含有链接框架的父框架或窗口中，如果含有该链接的框架不是嵌套的，则在浏览器全屏窗口中载入链接的文件，与_self 参数相同；_self 在同一框架或窗口中打开链接的文件，此参数为默认值，通常不必指定；_top 在当前的整个浏览器窗口中打开链接的文件，因此会删除所有框架；name 在指定的窗口或框架中打开网页。

在本项目的 admin-left.phtml 模板页中，通过超链接的 target 属性指定 mainwindow 框架，在该框架中打开链接的网页。

15.9　系 统 管 理

系统管理主要用于对系统公用信息进行设置及对管理员密码进行更改，通过该模块可以提高系统的可维护性。

实例 615	系统信息设置 光盘位置：光盘\MR\15\615	高级 趣味指数：★★★★☆

实例说明

明日科技图书网的系统信息设置主要用于设置商城 ID、企业账号、企业版权、联系方式和程序文件的保存路径等信息，系统信息设置的页面如图 15.46 所示。

图 15.46　系统信息设置页面

关键技术

在本实例中，关键之处不在于系统信息设置功能的实现方法，更重要的是它的操作。在进行系统信息设置时，必须设置最后 3 项内容，它们指定的是系统中图片、广告和试读文件的下载地址，如果设置的路径不正确，那么这 3 项内容将不能正确输出。

笔者在编写此项目时，将其路径设置为 http://127.0.0.1/MR/15/005/，这是在本地服务器运行的路径，如果将这个程序上传到互联网的服务器中，就必须修改这个路径，否则将找不到指定的图片和文件。

设计过程

系统信息设置功能由 admin-default.php 和 admin-default.phtml 两个文件组成。

（1）在 admin-default.php 文件中，完成数据库中存储的系统信息的输出和更新两项操作。首先，包含后台管理系统的头文件。然后，判断 form 表单中提交的 merid 值是否为空，如果不为空，则应用数据库管理类的 executeSQL()方法执行更新语句，更新数据库中存储的系统信息。最后，应用数据库管理类的 executeSQL()方法执行查询操作，查询数据库中存储的系统信息，并将其存储到模板变量中，其代码如下：

```
<?php
require_once 'admin-header.php';                                    //包含网站头文件
if (isset($_POST['merid']) && $_POST['merid'] != '') {              //判断表单提交的数据是否为空
    if (! $adminDB->executeSQL("update tb_system set merid='" . $_POST['merid'] . "', meracct='" . $_POST['meracct'] . "', readurl='" .
$_POST['readurl'] . "', bookimgurl='" . $_POST['bookimgurl'] . "', ggurl='" . $_POST['ggurl'] . "', bq='" . $_POST['bq'] . "' , address='" .
$_POST['address'] . "', tel='" . $_POST['tel'] . "', cz='" . $_POST['cz'] . "', email='" . $_POST['email'] . "', icp='" . $_POST['icp'] . "', qq='" . $_POST['qq'] .
"' where mark=1", $connID)) {
        echo "<script>alert('系统信息设置失败！ ');</script>";
    } else {
        echo "<script>alert('系统信息设置成功！ ');</script>";
    }
}
$system = $adminDB->executeSQL("select id, merid, meracct, readurl, bookimgurl, ggurl, bq, address, tel, cz, email, icp, qq from tb_system where
mark=1", $connID);
$smarty->assign('system', $system);                                 //将查询结果赋给模板变量
$smarty->display('admin-default.phtml');                            //指定模板页
require_once 'admin-footer.php';                                    //包含网站尾文件
```

（2）新建 admin-default.phtml 模板页，创建 form 表单，提交需要更新的系统信息，并且将模板变量中传递的系统信息作为表单元素的默认值。

秘笈心法

心法领悟 615：通过 JavaScript 脚本中的 chkinputsysteminfo()方法验证表单元素值。

在本实例的 admin-default.phtml 模板页中，调用 js/adminfun.js 文件中的 chkinputsysteminfo()方法对系统信息设置表单中提交的数据进行验证。

实例 616 更改管理员密码

光盘位置：光盘\MR\15\616

高级

趣味指数：★★★★☆

实例说明

为了便于网站管理者对网站进行管理，提高网站的安全性，在系统管理模块增加管理员密码修改功能，如图 15.47 所示。

图 15.47 管理员密码修改

关键技术

更改管理员密码的原理：用户在上述表单中输入原密码和新设定的密码，单击“更改”按钮将表单中的信息提交到 admin-changepwd.php 文件中对密码进行更改，在该文件中，首先判断用户输入的原密码是否正确，如



关键技术

在图书大类管理中，应用最多的就是数据库管理类中的 executeSQL()方法，通过该法执行 SQL 语句，完成数据库中数据的添加、更新、查询和删除操作，这也是所有后台管理系统中最为常用的管理手段。

设计过程

（1）添加图书大类信息由 admin-bigtype.phtml 和 admin-bigtype.php 两个文件组成。在 admin-bigtype.phtml 模板页中，创建 form 表单，将图书大类的名称提交到 admin-bigtype.php 文件中进行处理，并且创建添加图书大类信息和浏览图书大类信息的按钮，其关键代码如下：

```
<div style="width:98%; height:25px; text-align:left">
<input type="button" value="添加图书大类信息"   onclick="window.location.href='{util->baseUrl}/admin-bigtype.php'" />  
<input type="button" value="浏览图书大类信息" onclick="window.location.href='{util->baseUrl}/admin-listbigtype.php'"/>
</div>
```

在 admin-bigtype.php 文件中，获取表单中提交的图书大类名称，首先，应用数据库管理类中的 executeSQL()方法执行查询操作，判断在数据库中是否存在该类别，如果不存在，则应用数据库管理类中的 executeSQL()方法执行添加操作，完成图书大类的添加操作，其代码如下：

```
<?php
require_once 'admin-header.php';                                //包含网站头文件，在头文件中包含了配置文件
if (isset($_POST['typename']) && $_POST['typename'] != '') {              //判断类别名称是否为空
    if (! $adminDB->executeSQL("select id, typename from tb_bigtype where typename='" . trim($_POST['typename']) . "'", $connID)) {
        if (! $adminDB->executeSQL("insert into tb_bigtype(typename, addtime) values('" . trim($_POST['typename']) . "', '" . date('Y-m-d H:i:s') . "')",
$connID)) {
            echo "<scriipt>alert('类别添加失败！ ');</script>";
        } else {
            echo "<script>alert('类别添加成功！ ');</script>";
        }
    } else {
        echo "<script>alert('该类别已经添加！ ');</script>";
    }
}
$smarty->display('admin-bigtype.phtml');                              //指定模板页
require_once 'admin-footer.php';                                //包含尾文件
```

（2）浏览图书大类信息由 admin-listbigtype.php 和 admin-listbigtype.phtml 文件组成。在 admin-listbigtype.php 中，定义图书大类信息的删除、更新和查询 3 个方法，完成对图书大类信息的操作，其代码如下：

```
<?php
require_once 'admin-header.php';                                //包含头文件
if (isset($_GET['f']) && $_GET['f'] == 'del') {                         //判断是否是执行的删除操作
    if (! $adminDB->executeSQL("delete from tb_bigtype where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('类别删除失败！ ');</script>";
    }
}
$isShow = 'F';
if ((isset($_GET['f']) && $_GET['f'] == 'edit') || (isset($_POST['f']) && $_POST['f'] == 'edit')) {             //判断是否是执行的更新操作
    if (isset($_POST['f']) && $_POST['f'] == 'edit') {                                        //判断是否是执行的更新操作
        if (! $adminDB->executeSQL("update tb_bigtype set typename='" . $_POST['typename'] . "' where id='" . $_POST['id'] . "'", $connID)) {
            echo "<script>alert('类别更改失败！ ');</script>";
        } else {
            echo "<script>alert('类别更改成功！ ');</script>";
        }
    }
    if (isset($_GET['id'])) {                                    //获取数据的 ID
        $id = $_GET['id'];
    } else {
        $id = $_POST['id'];
    }
    $isShow = 'T';
    $bigtype = $adminDB->executeSQL("select id, typename from tb_bigtype where id='" . $id . "'", $connID);       //执行数据的查询操作
    $smarty->assign('bigtype', $bigtype);                       //将要重新编辑的信息赋给模板变量
}
$smarty->assign('isShow', $isShow);
```



（2）在图书小类展示模块中，主要显示出图书小类的名称、所属大类和小类添加时间，其运行效果如图 15.50 所示。

浏览图书小类信息			
类别名称	所属大类	添加时间	操作
C++/VC	程序设计	2009-05-21 10:23:37	
ASP	程序设计	2009-05-21 10:23:41	
PHP	程序设计	2009-05-21 10:26:14	
JAVA/JSP	程序设计	2009-05-21 10:26:16	

图 15.50　图书小类信息列表

单击图书小类信息列表页中的“编辑”按钮，将出现小类编辑表单，用户更改小类信息并提交表单后，表单内容将被提交到 admin-listsmalltype.php 文件中实现对图书小类信息的更改，其关键代码如下：

```
$isShow = 'F';
if ((isset($_GET['f']) && $_GET['f'] == 'edit') || (isset($_POST['f']) && $_POST['f'] == 'edit')) {
    $btypes = $adminDB->executeSQL("select id, typename from tb_bigtype order by addtime", $connID);    //获得大类信息
    for ($i = 0; $i < count($btypes); $i ++) {                                                             //将大类信息保存到数组中
        $arrayBtypeOption[$btypes[$i]['id']] = $btypes[$i]['typename'];
    }
    $smarty->assign('arrayBtypeOption', $arrayBtypeOption);
    if (isset($_POST['f']) && $_POST['f'] == 'edit') {                                                     //进行编辑操作
        if (! $adminDB->executeSQL("update tb_smalltype set typename='" . $_POST['typename'] . "', bigtypeid='" . $_POST['bigtypeid'] . "' where
id='" . $_POST['id'] . "'", $connID)) {
            echo "<script>alert('类别更改失败！ ');</script>";
        } else {
            echo "<script>alert('类别更改成功！ ');</script>";
        }
    }
    if (isset($_GET['id'])) {
        $id = $_GET['id'];
    } else {
        $id = $_POST['id'];
    }
    $isShow = 'T';
    $smalltype = $adminDB->executeSQL("select id, typename, bigtypeid from tb_smalltype where id='" . $id . "'", $connID);    //查询小类信息
    $smarty->assign('smalltype', $smalltype);
}
```

上述代码首先判断$_GET['f']数组元素是否已经设置，如果是，则说明用户单击“编辑”按钮，这时查询出所有大类名称，并使当前所编辑小类的大类名称处于选中状态，然后判断是否设置$_POST['f']数组元素的值，如果是，则说明已经提交小类信息更改表单，此时通过数据库管理类的 executeSQL()方法更改小类信息。

（3）如果管理员单击小类信息列表页中的“删除”按钮，则首先弹出一个提示框来提示用户是否确认删除，如果是，则通过数据库管理类的 executeSQL()方法执行 delete 语句删除指定的小类信息，其代码如下：

```
if (isset($_GET['f']) && $_GET['f'] == 'del') {
    if (! $adminDB->executeSQL("delete from tb_smalltype where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('类别删除失败！ ');</script>";                                              //执行删除
    }
}
```

秘笈心法

心法领悟 618：应用 isset()函数判断指定的变量是否存在。

isset()函数，检查变量是否被设置，如果设置则返回 TRUE，否则返回 FALSE。由于这是一个语言结构而非函数，因此它无法被“变量函数”调用，其语法如下：

```
bool isset ( mixed var [, mixed var [, ...]])
```

参数 var 为被检查的变量，可以有多个。

实例 619	出版社分类管理 光盘位置：光盘\MR\15\619	高级 趣味指数：★★★★☆

实例说明

图书不但可以按内容划分，还可以按其所属出版社进行分类，这样可以更加方便地引导读者选购自己信赖

的出版社所出版的图书。为使出版社分类更加形象、易于识别，在出版社分类中增加了出版社的 logo 图片信息，浏览出版社类别信息的页面效果如图 15.51 所示。

管理员：mr　　更改密码　退出登录
当前位置：明日网上书店 >> 管理中心 >> 出版社类别管理
添加出版社类别信息　浏览出版社类别信息
浏览出版社类别信息

出版社logo图片	出版社名称	添加时间	操作
	人民邮电出版社	2009-05-25 10:24:34	
	清华大学出版社	2009-05-25 10:24:53	
	机械工业出版社	2009-05-25 10:38:46	
	铁道出版社	2009-05-25 10:38:59	
	吉林人民出版社	2009-05-25 10:40:55	
	电子工业出版社	2009-05-25 10:41:13	

图 15.51　浏览出版社类别信息

关键技术

在出版社分类管理中，关键是出版社 logo 图标上传的实现方法，其上传应用的是 move_uploaded_file()函数。

（1）在执行上传之前，应用 is_dir()函数判断指定的文件夹是否存在，如果不存在则应用 mkdir()函数创建一个新文件夹。

（2）应用$_FILES[]全局数组获取上传图片的原始名称。

（3）定义上传文件在服务器文件夹下的存储名称，应用 date()获取时间戳，应用 mt_rand()函数定义随机数，通过 substr()函数获取原始文件的后缀名称，从而组成新的文件名称。

（4）定义上传文件在服务器中的存储路径。

（5）应用 move_uploaded_file()函数执行文件上传操作。

上述函数的详细讲解可以参考本书第 4 章的内容，这里不再赘述。

设计过程

（1）网站管理者在添加出版社名称并选择出版社 logo 图片后，单击“添加”按钮即可将出版社类别信息提交到 admin-pub.php 文件中进行处理，其代码如下：

```
if (isset($_POST['pubname']) && $_POST['pubname'] != '') {                    //如果提交了表单
    if (! $adminDB->executeSQL("select id, pubname from tb_pub where pubname='" . trim($_POST['pubname']) . "'", $connID)) {        //判断该出版社是否已经添加
        if (isset($_FILES["pubimg"]["name"]) && $_FILES["pubimg"]["name"] != "") {        //上传出版社 logo
            $dir = "./upfiles/bookimg";
            if (! is_dir($dir)) {
                mkdir($dir);
            }
            $upfilename = $_FILES["pubimg"]["name"];
            $filename = date("YmdHis") . mt_rand(1000, 9999) . substr($upfilename, strpos($upfilename, "."), strlen($upfilename) - strpos($upfilename, "."));                    //设置出版社保存文件名
            $address = $dir . "/" . $filename;
            @move_uploaded_file($_FILES["pubimg"]["tmp_name"], $address);            //执行上传操作
        } else {
            $filename = "";
        }
        if (! $adminDB->executeSQL("insert into tb_pub(pubname, pubimg, addtime) values('" . trim($_POST['pubname']) . "', '" . $filename . "', '" . date('Y-m-d H:i:s') . "')", $connID)) {                    //保存出版社类别信息
            echo "<script>alert('出版社类别添加失败！');</script>";
        } else {
            echo "<script>alert('出版社类别添加成功！');</script>";
        }
    } else {
        echo "<script>alert('该出版社类别已经添加！');</script>";
    }
}
```

上述代码使用函数 move_uploaded_file()上传出版社 logo，成功上传后，使用数据库管理类的 executeSQL()方法执行 insert 语句保存出版社类别信息。

（2）出版社类别信息列表页显示出版社 logo、出版社名称和类别添加时间。管理员单击出版社类别列表中的“编辑”超链接，将出现出版社类别更改表单，对出版社信息进行编辑，将表单内容提交到 admin-listpub.php 文件中进行处理，其代码如下：

```
$isShow = 'F';
if ((isset($_GET['f']) && $_GET['f'] == 'edit') || (isset($_POST['f']) && $_POST['f'] == 'edit')) {
    if (isset($_POST['f']) && $_POST['f'] == 'edit') {
        if (isset($_FILES["pubimg"]["name"]) && $_FILES["pubimg"]["name"] != "") {      //判断是否重新上传 logo
            $dir = "./upfiles/bookimg";
            if (! is_dir($dir)) {
                mkdir($dir);
            }
            $upfilename = $_FILES["pubimg"]["name"];
            $filename = date("YmdHis") . mt_rand(1000, 9999) . substr($upfilename, strpos($upfilename, "."), strlen($upfilename) -
strpos($upfilename, "."));
            ;
            $address = $dir . "/" . $filename;
            @move_uploaded_file($_FILES["pubimg"]["tmp_name"], $address);             //执行上传
        } else {
            $ptmp = $adminDB->executeSQL("select pubimg from tb_pub where id='" . trim($_POST['id']) . "'", $connID);
            $filename = $ptmp[0]['pubimg'];
        }
        if (! $adminDB->executeSQL("update tb_pub set pubname='" . $_POST['pubname'] . "', pubimg='" . $filename . "'   where id='" .
$_POST['id'] . "'", $connID)) {                                                      //更改出版社类别信息
            echo "<script>alert('出版社类别更改失败！ ');</script>";
        } else {
            echo "<script>alert('出版社类别更改成功！ ');</script>";
        }
    }
    if (isset($_GET['id'])) {
        $id = $_GET['id'];
    } else {
        $id = $_POST['id'];
    }
    $isShow = 'T';
    $pub = $adminDB->executeSQL("select id, pubname, pubimg from tb_pub where id='" . $id . "'", $connID);
    $smarty->assign('pub', $pub);
}
```

上述代码首先判断是否又重新上传出版社 logo，如果是，则重新上传，反之则不进行任何操作，然后使用数据库管理类的 executeSQL()方法执行 update 语句更改出版社类别信息。

（3）如果单击出版社类别列表“删除”按钮，并确认删除后，将通过如下代码删除指定的出版社类别信息：

```
if (isset($_GET['f']) && $_GET['f'] == 'del') {
    if (! $adminDB->executeSQL("delete from tb_pub where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('出版社类别删除失败！ ');</script>";
    }
}
```

秘笈心法

心法领悟 619：删除存储在数据表中的出版社类别信息。

这样删除的只是存储在数据表中的出版社类别信息，并不能将存储在服务器文件夹下的图片删除。如果要删除存储在服务器文件夹下的图片，必须在服务器文件夹下手动删除。

15.11 图 书 管 理

图书管理模块主要用于对图书信息和图书试读信息进行管理，本节将以实例的形式对明日科技图书网的图书信息管理模块进行讲解。

实例 620　图书信息管理

光盘位置：光盘\MR\15\620

高级

趣味指数：★★★★☆

实例说明

明日科技图书网的定位为图书产品的展示和销售，所以图书信息管理模块非常关键，该项目的图书信息管理模块主要包括对图书信息的添加、更改、删除和查询等操作。在图书信息添加页面中，添加的内容包括图书的名称、类别、页数、字数、作者、ISBN 号和价格等，其运行效果如图 15.52 所示。

图 15.52　图书信息添加

关键技术

在图书信息管理中，非常关键的一点是明确数据表的各个字段代表的含义，在明日科技图书网的前台中，在对图书信息进行分类时，根据的就是图书信息表中某个字段的值，例如，判断图书是否是新书、是否特价等。所以必须对图书信息表中各个字段了如指掌才能更好地完成判断操作。tb_bookinfo 图书信息表的说明如表 15.2 所示。

表 15.2　tb_bookinfo 图书信息表

字段名称	数据类型	字段大小	是否主键	说　明
id	int	11	主键	自动增长
bookname	varchar	50		大类名称
smalltypeid	datetime			添加时间
oldprice	float			图书原价
newprice	float			本站价格
bookimg	varchar	100		图书封面图片名称
isnew	bool			是否新书
addtime	datetime			添加时间
about	text			图书介绍
isspprice	bool			是否特价图书
browsertime	int	11		浏览次数



```
            if (! is_dir($dir)) {
                mkdir($dir);
            }
            $upfilename = $_FILES["bookimg"]["name"];
            $filename = date("YmdHis") . mt_rand(1000, 9999) . substr($upfilename, strpos($upfilename, "."), strlen($upfilename) -
strpos($upfilename, "."));
            ;
            $address = $dir . "/" . $filename;
            @move_uploaded_file($_FILES["bookimg"]["tmp_name"], $address);
        } else {
            $tmpbook = $adminDB->executeSQL("select bookimg from tb_bookinfo where id='" . $_POST['id'] . "'", $connID);
            $filename = $tmpbook[0]['bookimg'];
        }
        $pubtime = $_POST['pyear'] . '-' . $_POST['pmonth'] . '-00';
        ......//省略用户设置提交内容的代码，请参见本书附带光盘中源码
        if (! $adminDB->executeSQL("update tb_bookinfo set bookname='" . trim($_POST['bookname']) . "', smalltypeid='" .
trim($_POST['smalltypeid']) . "', pubid='" . trim($_POST['pubid']) . "', page='" . trim($_POST['page']) . "', zs='" . trim($_POST['zs']) . "', isbn='" .
trim($_POST['isbn']) . "', bc='" . trim($_POST['bc']) . "', writer='" . trim($_POST['writer']) . "', pubtime='" . $pubtime . "', oldprice='" .
trim($_POST['oldprice']) . "', newprice='" . trim($_POST['newprice']) . "', bookcc='" . trim($_POST['bookcc']) . "', bookids='" . trim($_POST['bookids']) .
"', bookimg='" . $filename . "', directory='" . trim($_POST['directory']) . "', about='" . trim($_POST['about']) . "', isnew='" . $isnew . "', issepprice='" .
$issepprice . "', ishotsell='" . $ishotsell . "', isterm='" . $isterm . "' ,ismrbooktj='" . $ismrbooktj . "' ,ishave='" . $ishave . "' where id='" . $_POST['id'] . "'",
$connID)) {                                                                                    //更改图书信息
            echo "<script>alert('图书信息更改失败 ！ ');</script>";
        } else {
            echo "<script>alert('图书信息更改成功 ！ ');</script>";
        }
    }
    if (isset($_POST['id'])) {
        $id = $_POST['id'];
    } else {
        $id = $_GET['id'];
    }
    $isEdit = 'T';
    $book = $adminDB->executeSQL("select id, bookname, smalltypeid, pubid, page, zs, isbn, bc, writer, pubtime, oldprice, newprice, bookcc, bookids,
bookimg, directory, about, isnew, issepprice, ishotsell, isterm ,ismrbooktj ,ishave, addtime, browsertime from tb_bookinfo where id='" . $id . "' ",
$connID);     //查询图书信息
    $smarty->assign('book', $book);
    $smarty->assign('pyear', substr($book[0]['pubtime'], 0, 4));
    $smarty->assign('pmonth', substr($book[0]['pubtime'], 5, 2));
}
```

在上述代码中，定义图书信息添加和更新两种操作。首先判断$_POST['change']数组元素的值，如果该值为 F，则进行添加操作，通过数据库管理类的 executeSQL()方法执行 insert 语句将图书信息保存到数据库中，同时使用函数 move_uploaded_file()将图书封面上传到服务器中；如果$_POST['change']的值为 T，则执行图书信息的更改操作，在更改图书信息时，首先判断是否重新选择图书封面，如果是，则重新上传图书封面，然后执行 update 语句对图书信息进行更新。

（2）图书信息列表主要对图书名称、出版社、作者和价格等进行分页显示，其运行效果如图 15.53 所示。

浏览图书信息

图书名称	出版社	作者	市场价（元）	会员价（元）	操作
《PHP开发典型模块大全》	人民邮电出版社	邹天思 潘凯华 孙鹏	79.00	79.00	
《Visual C++开发经验技巧宝典》	人民邮电出版社	明日科技	89.00	89.00	
《Visual C++程序设计自学手册》	人民邮电出版社	明日科技 宋坤 刘锐宁	52.00	52.00	
《SQL Server 2005开发技术大全》	人民邮电出版社	明日科技	79.00	79.00	
《C#程序设计标准教程》	人民邮电出版社	明日科技 王小科 吕双	59.00	59.00	
《ASP.NET网络编程标准教程》	人民邮电出版社	张跃廷 丁红兵	55.00	55.00	

图 15.53　图书信息列表

通过图书信息列表页可以实现对图书信息的更改和删除操作，在讲解图书添加操作中已经介绍了图书信息更改的实现方法，这里不再赘述。当单击图书信息列表中的“删除”按钮时，首先弹出提示对话框提示用户是否确认删除，如果是，则通过如下代码执行对图书信息的删除操作：

```
if (isset($_GET['f']) && $_GET['f'] == 'del') {
    if (! $adminDB->executeSQL("delete from tb_bookinfo where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('图书信息删除失败！ ');</script>";                    //执行删除操作
    }
}
```

（3）与其他管理模块相比，图书信息量较大，为方便对图书信息进行管理，在图书信息管理模块中增加了图书信息查询功能，其运行效果如图 15.54 所示。

图 15.54 图书信息查询

首先在文本框中输入要查询的关键字，然后单击“查询”按钮，这时查询表单的关键字将被提交到 admin-searchbook.php 文件中，在该文件完成对图书信息的查询操作，并且将查询结果赋给模板变量，在模板页 admin-searchbook.phtml 中输出查询结果，其代码如下：

```
$isFind = 'F';
if (isset($_POST['bookname']) && $_POST['bookname'] != '' || isset($_GET['bookname']) && $_GET['bookname'] != '') {    //判断是否提交了表单
    if (isset($_POST['bookname'])) {                                                                                    //获取要查询的图书名称
        $bookname = $_POST['bookname'];
    } else {
        $bookname = $_GET['bookname'];
    }
    $books = $adminDB->executeSQL("select tb_bookinfo.id , tb_bookinfo.oldprice, tb_bookinfo.newprice, tb_bookinfo.bookname,
tb_bookinfo.writer,  tb_pub.pubname from tb_bookinfo, tb_pub where tb_bookinfo.pubid = tb_pub.id and bookname like '%" . $bookname . "%'  order
by tb_bookinfo.addtime desc", $connID);                                                                                 //执行模糊查询
    $smarty->assign('books', $books);
    $isFind = 'T';
    $smarty->assign('bookname', $bookname);
}
```

上述代码首先使用函数 isset()判断$_POST['bookname']数组元素是否已经被设置，如果是，则使用数据库管理类的 executeSQL()方法执行查询操作。

秘笈心法

心法领悟 620：通过 like 关键字执行模糊查询，查询图书的名称。

在本实例的图书查询中，应用 like 关键字执行多表模糊查询，查询图书的名称，并且将查询结果返回到模板变量中。

实例 621 图书试读管理

光盘位置：光盘\MR\15\621

高级

趣味指数：★★★★☆

实例说明

为了让读者选择一本真正适合自己的图书，和吸引更多读者购买，在制作明日科技图书网时增加了图书试读模块，这样读者在购买图书前就可以大概了解所购买图书的内容，非常人性化。在图书试读信息添加页面中包括图书类别下拉列表框、图书名称选择下拉列表框和试读文件名称输入文本框，其运行效果如图 15.55 所示。

图 15.55 添加图书试读信息页面

关键技术

在添加图书试读信息的模板页 admin-read.phtml 中，通过 Smarty 模板中的{ldelim} {rdelim}标签直接在模板页中编写 JavaScript 脚本，应用 jQuery 技术实现下拉菜单的三级联动，其关键代码如下：

```
<script src="{util->baseUrl}/js/jquery.js"></script>
```



浏览图书试读信息

图书名称	试读文件名	添加时间	操作
《Visual C++开发经验技巧宝典》	1.doc	2009-07-03 17:45:13	
《Visual C++程序设计自学手册》	1.doc	2009-07-03 17:44:39	

共 2 本　每页显示 20 本　第 1 页/共 1 页　　首页　上一页　下一页　尾页

图 15.56　图书试读信息列表

单击试读信息列表的“编辑”图标将出现试读信息更改表单，在该表单中修改试读图书的内容，将表单数据提交到 admin-listread.php 页面中进行修改，其代码如下：

```
if ((isset($_GET['f']) && $_GET['f'] == 'edit') || (isset($_POST['f']) && $_POST['f'] == 'edit')) {                //判断是否提交了表单
    if (isset($_POST['f']) && $_POST['f'] == 'edit') {
        if (! $adminDB->executeSQL("update tb_read set bookinfoid='" . $_POST['bookid'] . "', filename='" . $_POST['filename'] . "'    where id='" .
$_POST['id'] . "'", $connID)) {      //执行更改操作
            echo "<script>alert('试读更改失败！ ');</script>";
        } else {
            echo "<script>alert('试读更改成功！ ');</script>";
        }
    }
    if (isset($_GET['id'])) {
        $id = $_GET['id'];
    } else {
        $id = $_POST['id'];
    }
    $isShow = 'T';
    $read = $adminDB->executeSQL("select tb_read.id, tb_read.filename, tb_bookinfo.bookname, tb_read.addtime from tb_read, tb_bookinfo where
tb_read.bookinfoid = tb_bookinfo.id and tb_read.id='" . $id . "'", $connID);   //查询试读信息
    $smarty->assign('read', $read);
}
```

单击试读信息的“删除”图标，会弹出是否确认删除的对话框，如果选择确认删除，则执行如下代码实现试读信息的删除：

```
if (isset($_GET['f']) && $_GET['f'] == 'del') {
    if (! $adminDB->executeSQL("delete from tb_read where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('试读删除失败！ ');</script>";    //执行删除试读操作
    }
}
```

秘笈心法

心法领悟 621：分页功能的实现。

在本实例中，试读信息的分页输出应用的是分页类中的 pageData()方法，将分页查询结果赋给模板变量，在模板页中创建分页超链接，完成试读信息的分页输出。

15.12　用　户　管　理

用户管理模块实现的功能包括对网站注册用户的管理和对用户反馈信息的管理，本节将讲解这两个功能的实现过程。

实例 622	用户管理 光盘位置：光盘\MR\15\622	高级 趣味指数：★★★★

实例说明

在用户管理模块中对网站注册用户进行分页输出，输出内容包括用户昵称、真实姓名、性别和联系电话等，其运行效果如图 15.57 所示。



实例 623 用户反馈管理

光盘位置：光盘\MR\15\623

高级

趣味指数：★★★★☆

实例说明

用户反馈管理模块输出用户对网站本身、网站中的某本图书，或者其他方面的一些建议、意见等，其输出效果如图 15.58 所示。

浏览用户反馈信息

主题	用户	发表时间	操作
程序开发经验交流	mr	2010-08-27 02:35:37	
我的建议	mr	2010-05-13 07:52:42	
用户意见	2000	2009-08-13 12:28:10	

共 3 条 每页显示 20 条 第 1 页/共 1 页 首页 上一页 下一页 尾页

查看用户反馈详细信息

用户昵称：mr 发表时间：2010-05-13 07:52:42

发表时间：我的建议

我的建议

图 15.58 浏览用户反馈信息

关键技术

本实例实现的功能与实例 622 是相同的，唯一的不同之处是本实例操作的是 tb_feedback 数据表，而用户管理实例操作的是 tb_user 表。

设计过程

（1）浏览用户反馈信息由 admin-listuserfeedback.php 和 admin-listuserfeedback.phtml 两个文件组成。在 admin-listuserfeedback.php 文件中，首先，判断指定的分页变量是否存在。然后，根据$_GET[]获取的变量值进行判断，执行用户反馈信息的删除操作、数据的分页输出或者查看指定用户反馈的详细信息。最后指定模板页，其代码如下：

```
<?php
require_once 'admin-header.php';                              //包含头文件
if (! isset($_GET['page']) || $_GET['page'] == '') {           //判断分页变量是否存在
    $page = '1';
} else {
    $page = $_GET['page'];
}
if (isset($_GET['f']) && $_GET['f'] == 'del') {               //判断是否执行删除操作
    if (! $adminDB->executeSQL("delete from tb_feedback where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('用户反馈信息删除失败！');</script>";
    }
}
$sql = "select id, usernc, title, addtime from tb_feedback order by addtime desc";
$feedbacks = $pageDB->pageData($sql, $connID, 20, $page);     //执行分页查询
$smarty->assign('feedbacks', $feedbacks);
$isShow = 'F';
if (isset($_GET['f']) && $_GET['f'] == 'edit') {              //判断是否执行详细信息查询操作
    $userfeed = $adminDB->executeSQL("select id, usernc, title, content, addtime from tb_feedback where id='" . $_GET['id'] . "'", $connID);
    $isShow = 'T';
    $smarty->assign('userfeed', $userfeed);                   //将查询结果赋给模板变量
}
$smarty->assign('isShow', $isShow);
$smarty->display('admin-listuserfeedback.phtml');             //指定模板页
require_once 'admin-footer.php';                              //包含尾文件
```

（2）新建 admin-listuserfeedback.phtml 模板页。首先，通过 section 语句循环输出用户反馈信息，并且创建查看指定用户反馈详细信息和删除指定用户的超链接。然后，定义分页超链接，完成数据的分页输出。最后，



续表

字 段 名 称	数 据 类 型	字 段 大 小	是 否 主 键	说　　明
numstr	varchar	100		订购图书数量组成字符串
goodsprice	float			图书价格
yjprice	float			邮寄价格
totalprice	float			总价格
rectype	varchar	20		收货方式
paytype	varchar	20		支付方式
addtime	datetime			订购时间
isfk	bool			是否付款
isfh	bool			是否发货
issh	bool			是否收货
isqx	bool			是否被用户取消订购

设计过程

用户订单管理首先完成用户订单信息的分页输出，然后执行订单支付状态的更改、订单信息的查看和删除等操作。

（1）在 admin-listorder.php 文件中，首先，包含头文件，根据传递的查询字符串 page 的值并通过数据库分页类的 pageData()方法实现订单信息的分页输出。然后，通过数据库管理类的 executeSQL()方法执行 update 语句实现订单状态的更改。最后，通过数据库管理类的 executeSQL()方法执行 select 语句，查询指定订单的详细信息，并且将查询结果在模板页中输出。admin-listorder.php 文件的代码如下：

```
<?php
require_once 'admin-header.php';                                    //包含头文件
if (! isset($_GET['page']) || $_GET['page'] == '') {                //判断分页变量的值
    $page = '1';
} else {
    $page = $_GET['page'];
}
if (isset($_GET['f']) && $_GET['f'] == 'del') {                     //判断是否执行删除操作
    if (! $adminDB->executeSQL("delete from tb_order where id='" . $_GET['id'] . "'", $connID)) {
        echo "<script>alert('订单信息删除失败！ ');</script>";
    }
}
if (isset($_GET['c']) && $_GET['c'] != '') {                        //更改订单状态
    $c = $_GET['c'];
    if ($c == 'isfk') {                                             //判断是否已经付款
        $adminDB->executeSQL("update tb_order set isfk = !isfk where id='" . $_GET['id'] . "'", $connID);
    } elseif ($c == 'isfh') {                                       //判断是否已经发货
        $adminDB->executeSQL("update tb_order set isfh = !isfh where id='" . $_GET['id'] . "'", $connID);
    } elseif ($c == 'issh') {                                       //判断是否已经收货
        $adminDB->executeSQL("update tb_order set issh = !issh where id='" . $_GET['id'] . "'", $connID);
    }
}
$sql = "select id, orderno, username, tel, goodsprice, yjprice, totalprice, isfk, isfh, issh, isqx from tb_order order by addtime desc";
//定义 SQL 语句
$orders = $pageDB->pageData($sql, $connID, 20, $page);              //执行分页查询
$smarty->assign('orders', $orders);                                 //将查询结果赋给模板变量
$isShow = 'F';
if (isset($_GET['f']) && $_GET['f'] == 'edit') {                    //判断是否执行查看订单详细信息的操作
    $order = $adminDB->executeSQL("select id, orderno, username, address, sex, yb, tel, idstr, numstr, rectype, paytype, goodsprice, yjprice, totalprice,
addtime, isfk, isfh, issh from tb_order where id='" . $_GET['id'] . "'", $connID);    //执行查询
    $arrayIds = explode('@', $order[0]['idstr']);
    $arrayNums = explode('@', $order[0]['numstr']);
    $arrayCarInfos = array();
    $totalPrice = 0;
    for ($i = 0; $i < count($arrayIds); $i ++) {
```

```
            $bookid = $arrayIds[$i];
            if ($bookid != '') {
                $tmpArray = array();
                $bookinfo = $adminDB->executeSQL("select id, bookname ,oldprice, newprice from tb_bookinfo where id='" . $bookid . "'", $connID);
                $tmpArray['id'] = $bookinfo[0]['id'];
                $tmpArray['bookname'] = $bookinfo[0]['bookname'];
                $tmpArray['oldprice'] = $bookinfo[0]['oldprice'];
                $tmpArray['newprice'] = $bookinfo[0]['newprice'];
                $tmpArray['num'] = $arrayNums[$i];
                $tmpArray['smallTotalPrice'] = $bookinfo[0]['newprice'] * $arrayNums[$i];
                $totalPrice += $tmpArray['smallTotalPrice'];
                array_push($arrayCarInfos, $tmpArray);
            }
        }
        $smarty->assign('arrayCarInfos', $arrayCarInfos);                    //将生成的订单详细信息赋给模板变量
        $smarty->assign('order', $order);
        $isShow = 'T';
}
$smarty->assign('isShow', $isShow);                                          //通过该变量值判断是否输出订单详细信息
$smarty->display('admin-listorder.phtml');                                   //指定模板页
require_once 'admin-footer.php';                                             //包含尾文件
```

（2）在 admin-listorder.phtml 模板页中，首先，通过 section 语句循环输出订单信息，创建复选框更改用户订单的状态，创建查看订单详细信息和删除指定订单的超链接。然后，创建数据分页超链接，完成订单信息的分页输出。最后，输出模板变量中存储的指定订单的详细信息。其关键代码如下：

```
<div style="border-right:1px solid #006D84;{if $orders.data[oID].isqx==1}padding-top:4px{/if}">
{if $orders.data[oID].isqx==1}--已取消--{else}付款：<input type="checkbox" name="isfk" {if $orders.data[oID].isfk==1}checked{/if}
onclick="window.location.href='admin-listorder.php?c=isfk&id={$orders.data[oID].id}'" />  
发货：<input type="checkbox" name="isfh" {if $orders.data[oID].isfh==1}checked{/if}
onclick="window.location.href='admin-listorder.php?c=isfh&id= {$orders.data[oID].id}'">  
收货：<input type="checkbox" name="issh" {if $orders.data[oID].issh==1}checked{/if}
onclick="window.location.href='admin-listorder.php?c=issh&id= {$orders.data[oID].id}'">
{/if}
</div>
<div style="width:9%; height:20px; float:left; padding-top:2px">
<a href="{util->baseUrl}/admin-listorder.php?f=edit&id={$orders.data[oID].id}"><img src="{util->baseUrl}/img/edit.gif" border="0"/></a> 
<a href="javascript:if(window.confirm('确定删除？')==true){ldelim}window.location.href='{util->baseUrl}/admin-listorder.php?f=del&id=
{$orders.data[oID].id}';{rdelim}"><img src="{util->baseUrl}/img/del.gif" border="0"/></a>
</div>
```

秘笈心法

心法领悟 624：通过复选框完成订单状态的更改。

在 admin-listorder.phtml 模板页中，创建复选框，通过 onclick 事件调用 JavaScript 脚本，在 JavaScript 脚本中，应用 window 对象的 location 方法调用 admin-listorder.php 文件，并传递两个参数 c 和 id，在 admin-listorder.php 文件中完成订单状态的更改操作。

15.14　通用信息管理

在通用信息管理模块可实现对新闻公告、网上畅销书讯和明日图书排行管理，本节将讲解其功能的具体实现方法。

实例 625	新闻公告管理 光盘位置：光盘\MR\15\625	高级 趣味指数：★★★★☆

实例说明

在新闻公告管理模块中，可实现新闻公告的添加、浏览、更新和删除操作。其中在添加新闻公告功能中，

通过 Fckeditor 文本编辑器编辑公告的内容，其运行效果如图 15.60 所示

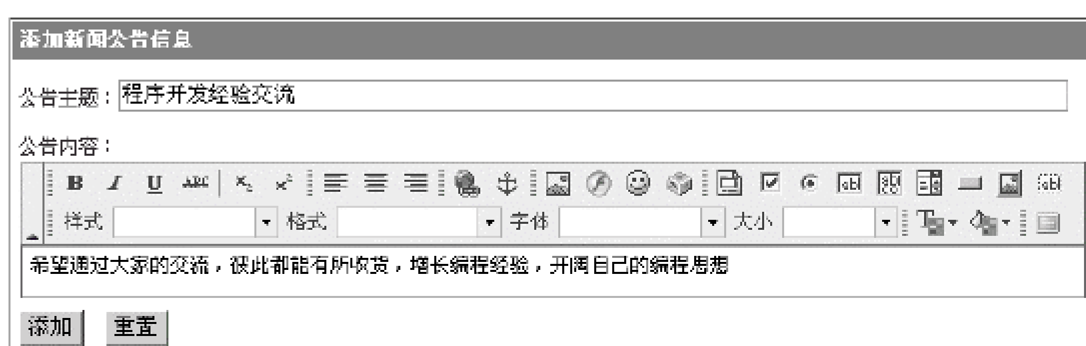


图 15.60 添加新闻公告信息

关键技术

在本实例中，关键是如何在 Smarty 模板页中嵌入 Fckeditor 文本编辑器，其应用的是 Util 类中的 editor()方法。editor()方法的语法如下：

```
function editor ($name, $value, $width = '100%', $height = '200'){
        require_once 'library/fckeditor/fckeditor.php';
        $arrayIni = $this->arrayIni;
        $oFCKeditor = new FCKeditor($name);
        $oFCKeditor->BasePath = $arrayIni['baseUrl'] . '/library/fckeditor/';
        $oFCKeditor->Width = $width;
        $oFCKeditor->Height = $height;
        $oFCKeditor->ToolbarSet = 'Default';
        $oFCKeditor->Value = $value;
        $oFCKeditor->Create();
    }
```

在 lzh.inc.php 文件中，应用 Smarty 模板中的 register_object()方法注册一个模板对象，其关键代码如下：

```
$util =    new Util();
$smarty->register_object('util', $util, null, false);
```

在模板页中通过注册的模板对象调用 Util 类中的 editor()方法完成 Fckeditor 文本编辑器的嵌入操作，其关键代码如下：

```
{if $isEdit=="F"}
{util->editor p1 ='content' p2 =''}<br /><br />
{else}
    {util->editor p1 ='content' p2 =$info[0].content}<br /><br />
    <input type="hidden" name="id" value="{$info[0].id}" />
{/if}
```

设计过程

（1）新闻公告的添加由 admin-tell.php 和 admin-tell.phtml 两个文件组成。在 admin-tell.phtml 模板页中创建 form 表单，载入 Fckeditor 文本编辑器，将新闻公告提交到 admin-tell.php 文件，完成新闻公告的添加操作。在 admin-tell.php 文件中，调用数据库管理类中的 executeSQL()方法完成新闻公告的添加、更新和查询操作，其关键代码如下：

```
<?php
require_once 'admin-header.php';                                                  //包含头文件
if(isset($_POST['title']) && $_POST['title']!=''&&!isset($_POST['id'])){          //判断提交的 ID 和标题是否为空
 if(!$adminDB->executeSQL("select id, title from tb_tell where title='".trim($_POST['title'])."'", $connID)){
        if(!$adminDB->executeSQL("insert into tb_tell(title, content, addtime) values('".trim($_POST['title']).'',
''.trim($_POST['content']).'','".date('Y-m-d H:i:s')."')", $connID)){
            echo "<scriipt>alert('公告添加失败！ ');</script>";
        }else{
            echo "<script>alert('公告添加成功！ ');</script>";
        }
    }else {
        echo "<script>alert('该公告已经添加！ ');</script>";
    }
}
$isEdit = 'F';
if(isset($_GET['f']) && $_GET['f']=='edit' || isset($_POST['id']) && $_POST['id']!=''){              //判断是否执行更新操作
```



附　录——字母索引

字母 A 关键字索引



续表

字母 D 关键字索引



续表

字母 F 关键字索引

续表





字母 J 关键字索引

字母 L 关键字索引

字母 M 关键字索引



续表

字母 N 关键字索引

字母 O 关键字索引



续表

字母 R 关键字索引

续表





字母 W 关键字索引

字母 X 关键字索引

其他关键字索引